Enhancing IoT Systems by Exploiting Opportunistically Collected Information from Communication Networks

Mikkelsen, Lars Møller

DOI (link to publication from Publisher):
10.5278/VBN.PHD.TECH.00008

Publication date:
2017

Document Version
Publisher's PDF, also known as Version of record

ENHANCING IOT SYSTEMS BY EXPLOITING OPPORTUNISTICALLY COLLECTED INFORMATION FROM COMMUNICATION NETWORKS

BY
LARS MØLLER MIKKELSEN

DISSERTATION SUBMITTED 2017

AALBORG UNIVERSITY
DENMARK

Enhancing IoT Systems by Exploiting Opportunistically Collected Information from Communication Networks

Ph.D. Dissertation

Lars Møller Mikkelsen

Aalborg University
Department of Electronic Systems
Fredrik Bajers Vej 7A
DK-9220 Aaolborg Øst

Dissertation submitted:	March, 2017
PhD supervisor:	Associate Professor Tatiana K. Madsen Aalborg University
Assistant PhD supervisor:	Professor Hans-Peter Schwefel Aalborg University
PhD committee:	Associate Prof. Troels Bundgaard Sørensen (Chair.) Department of Electronic Systems Aalborg University
	Professor Nöel Crespi Institut Mines-Télécom Ecole nationale supérieure des Télécommunications
	Professor Laurent Schumacher Faculty of Computer Science University of Namur
PhD Series:	Technical Faculty of IT and Design, Aalborg University

ISSN (online): 2446-1628
ISBN (online): 978-87-7112-938-0

Published by:
Aalborg University Press
Skjernvej 4A, 2nd floor
DK – 9220 Aalborg Ø
Phone: +45 99407140
aauf@forlag.aau.dk
forlag.aau.dk

Printed in Denmark by Rosendahls, 2017

Abstract

In this thesis, we investigate how to enhance internet of things (IoT) systems by utilizing information about network performance and about number of people in a limited area. Network performance information is relevant for distributed IoT systems as they are affected by the performance of the network between their components. Knowing about the performance of the network allows these systems to optimize scheduling of communication. People count information is relevant for IoT systems because they make out the link between the digital and physical world interacting with users. Knowing about the number of users in an area translates to into the load or availability of the system. With these information types available, IoT systems can be made truly smart and adaptable to the context they operate in, and not just operate based on a predefined set of rules or threshold values. The main challenge in obtaining these information types is how to measure and collect them. This is because they are highly dynamic as they are influenced by the current situation at the location where they are measured. In this work, we propose how to obtain these parameters cost effectively by opportunistically collecting them using automated approaches. Specifically, network performance information is collected using crowd sourcing measurements and people count information is obtained as estimations based on passively collected WiFi probes. Furthermore, we propose how to exploit the collected information types in a real IoT system.

The thesis consists of three main contributions. In the first contribution, we present NetMap, a system we developed for performing and collecting transport layer end-to-end network performance measurements. NetMap exploits the ubiquitous presence of smartphones to collect many measurements at low cost, which are used for populating a network performance map with real network measurements. We focus on the network performance between end user device and back end system, including the 3G and 4G cellular connections and their influence on the application traffic. Specifically, we look at the metrics round-trip time and achievable throughput in estimating end-to-end transport layer network performance, but we also include received power measurements. In doing this we perform an evaluation of two methods for

estimating achievable throughput with the goal of reducing data usage in the estimation. We also verify received power measurements performed using consumer smartphones, by comparing with measurements performed with professional measurement equipment. Finally, we evaluate the impact of interpolating measurements when estimating mean performance parameters in sparsely populated areas in a network performance map. We show that there is a clear benefit from interpolating round-trip times and received power measurements.

In the second contribution, we focus on people count estimation, and specifically on estimating bus occupancy based on passively collected WiFi probes. We create a system for passively collecting WiFi probes from sensors placed on buses. Based on the collected probes we estimate the number of devices on the bus using a maximum likelihood estimator (MLE). Based on the MLE we estimate the number of people on the bus by assuming probabilities of a person carrying a certain number of devices. This system shows that it is feasible to obtain estimates for number of people on a bus in a cost-effective and automated manner. Finally, we design a quality indicator to attach to the estimated people count. The quality indicator is based on the estimator for number of people and on the maximum number of passengers on the bus. The quality indicator can be used for applications to evaluate the information quality and modify their behavior accordingly.

In the third contribution, we focus how IoT systems can exploit the previously described information types, both in enhancing the system and in improving service discovery and selection. To do this we create a smart parking system that supports users in finding and paying for parking. The smart parking system is developed as a use case demonstrating the MOBiNET platform and its functionalities. The parking system utilizes the MOBiNET platform to automatically discover and select parking information services via the location based service discovery. By using the MOBiNET platform we demonstrate how the system automatically can expand the coverage in terms of information sources and user reach.

Resumé

I denne afhandling undersøger vi hvordan internet of things (IoT) kan forbedres ved at udnytte information om netværks ydelse og om antallet af personer i et afgrænset geografisk område. Information on netværks ydelse er relevant for distribuerede IoT systemer da de er påvirket af ydelsen af det netværk imellem deres komponenter. Ved at kende netværkets ydelse er disse systemer i stand til at optimere planlægningen af deres kommunikation. Antallet af personer er relevant for IoT systemer fordi de udgør bindeleddet mellem den digitale og den fysiske verden, hvor de interagerer med brugere. At kende antallet af personer i et område kan oversættes til belastningen eller tilgængeligheden af systemet. Med disse informationstyper tilgængelige kan IoT systemer bliver gjort rigtigt smarte og tilpasningsdygtige til den kontekst de opererer i, og ikke bare operere baseret på et foruddefineret sæt af regler eller grænseværdier. Hovedudfordringen i at gøre disse informations typer tilgængelige er hvordan de måles og indsamles. Dette er er fordi de er meget dynamiske da de påvirkes af den situation på det sted hvor de måles. I dette værk foreslår vi hvordan vi kan skaffe disse parametre på en omkostningseffektiv måde ved at indsamle dem på opportunistisk og automatiseret vis. Specifikt indsamler vi netværksydelsesinformation ved hjælp af crowd sourcing af målinger, og vi indsamler information om person antal som et estimat baseret på passivt indsamlet WiFi prober. Endvidere foreslår vi hvordan disse informationstyper kan udnyttes i et rigtigt IoT system.

Denne afhandling består af tre hoved bidrag. I det første bidrag præsenterer vi NetMap, et system vi har udviklet til at foretage og indsamle transport lags end-to-end netværksydelsesmålinger. NetMap udnytter de allestedsnærværende smartphones til at indsamle mange målinger ved lave omkostninger, som bliver brugt til at udfylde et netværksydelseskort med rigtige netværksmålinger. Vi fokuserer på netværksydelsen mellem brugerenheder og server systemer, hvilket inkluderer 3G og 4G mobilnetværksforbindelser og deres indflydelse på applikationstrafik. Vi kigger specifikt på parametrene round-trip time og opnåelig throughput for at estimere end-to-end netværksydelse på transportlaget, men vi inkluderer også received power målinger. Vi evaluerer to metoder til at estimere opnåelig throughput med

det mål at reducere dataforbruget i estimeringen. Vi verificerer også received power målinger udført med forbruger smartphones, ved at sammenligne med målinger foretaget med professionelt måleudstyr. Til sidst evaluerer vi indflydelsen af at interpolere målinger når middelværdien estimeres i områder som er tyndt dækket af målinger, i netværksydelseskort. Vi viser at der er en klar gavn af at interpolere round-trip time målinger og received power målinger.

I det andet bidrag fokuserer vi på at estimere personantal, og specifikt at estimere antal personer på en bus baseret på passivt indsamlet WiFi prober. Vi laver et system til passivt at indsamle WiFi prober fra sensorer placeret på busser. Baseret på de indsamlede prober estimerer vi antallet af enheder på bussen ved at bruge en maximum likelihood estimator (MLE). Baseret på denne MLE estimerer vi antallet af personer på bussen ved at antage sandsynligheden for at en person bærer et bestemt antal enheder. Dette system viser at det er muligt at nå til antallet af personer på en bus både omkostningseffektivt og automatisk. Til sidst designer vi en kvalitetsindikator som kan vedhæftes det estimerede antal personer. Kvalitetsindikatoren er baseret på en kombination af estimeringen af antallet af personer og den højeste kapacitet på bussen. Kvalitetsindikatoren kan bruges af applikationer til at evaluere informationskvaliteten og ændre måden de bruger informationen på afhængigt af kvaliteten.

I det tredje bidrag fokuserer vi på hvordan IoT systemer kan udnytte de tidligere beskrevet informationstyper, både i forbindelse med at forbedre systemet og i forbindelse med at forbedre service discovery og service valg. For at gøre dette lavede vi et smart parkeringssystem som hjælper brugerne med at finde og betale for parkering. Dette smarte parkeringssystem er udviklet som en use case der skal demonstrere MOBiNET platformen of dennes funktioner. Parkeringssystemet benytter MOBiNET platformen til at automatiske foretage service discovery og vælge parkeringsinformationsservices ved hjælp af lokationsbasseret service discovery. Ved at benytte MOBiNET platformen demonstrerer vi hvordan systemet automatisk kan udvides til at dække flere informationskilder og flere brugere.

Contents

Abstract **iii**

Resumé **v**

Thesis Details **xiii**

Acknowledgments **xv**

I Introduction **1**

1 Motivation and Problem Description **3**

1.1 Thesis Context 3

1.2 Background and Motivation 4

1.3 Problem Area Delimitation 7

1.4 Problem Statement 8

1.5 Outline and Overview of Contributions 9

References 10

2 Related Work **11**

2.1 End-to-End Network Performance Estimation 11

2.1.1 Network Performance Map (NPM) 11

2.1.2 Network Performance Metrics 12

2.1.3 Measurement Methods 13

2.2 Crowd Sourcing Network Performance Measurements Using Smartphones 14

2.3 People Counting Techniques 15

2.4 IoT Platforms and Service Discovery 16

References 18

3 Crowd Sourcing Tool for Network Performance Measurements **23**

3.1 Problem Description and Delimitations 23

3.2 Realization of System for Cellular Network Performance Measurements Using Crowd Sourcing 24
3.3 Estimating Network Performance Metrics in a Crowd Sourcing Setting 26
3.3.1 Evaluation of Methods for Estimating Achievable Throughput 27
3.3.2 Evaluation of Received Power Measurements Obtained from Smartphones 28
3.4 Dealing with Heterogeneous Measurement Distribution 29
3.5 Conclusion & Outlook 30
3.6 Included Papers 32
References 34

4 Estimating Bus Occupancy Based on WiFi Probes 37
4.1 Problem Description and Delimitations 37
4.2 WiFi Probe Collection System 39
4.3 Obtaining People Count 40
4.3.1 Estimating Number of Devices 40
4.3.2 Estimating Number of People 41
4.4 Indicating Quality of Estimator 42
4.5 Conclusion & Outlook 43
4.6 Included Papers 44
References 45

5 Enhancing IoT Systems by Including Contextual Information 47
5.1 Problem Description and Delimitations 47
5.2 Intelligent Parking Assistant System 49
5.3 Envisioned Information Exploitation 52
5.3.1 Exploiting People Count Information 52
5.3.2 Exploiting Network Performance Information 53
5.4 Conclusion and Outlook 54
5.5 Included Papers 55
References 56

6 Conclusion and Outlook 59
6.1 Conclusion 59
6.2 Outlook 61

II Papers 63

A NetMap - Creating a Map of Application Layer QoS Metrics of Mobile Networks Using Crowd Sourcing 65
A.1 Introduction 67

A.1.1 Advantages of NetMap 68
A.1.2 Existing Solutions 68
A.1.3 Measurement Approach in NetMap 69
A.1.4 Requirements to NetMap 70
A.2 General System Functionality/Architecture 70
A.2.1 End User Controls 71
A.2.2 End User/Crowd Motivation 72
A.3 Metrics and Measurement Methods 72
A.3.1 Application Layer QoS Metrics 72
A.3.2 Physical Layer Connection Metrics 74
A.3.3 Notes Regarding NetMap Measurements 75
A.4 Examples of Results 76
A.5 Conclusion 77
References 79

B Performance Evaluation of Methods for Estimating Achievable Throughput on Cellular Connections 81
B.1 Introduction 83
B.2 What Will Be Measured 85
B.3 Measurement Setup and Methodology 86
B.4 Bulk Transfer Capacity (BTC) 87
B.4.1 BTC Setup 88
B.4.2 BTC Evaluation 89
B.5 Trains Of Packet-Pair (TOPP) 90
B.5.1 BTC Alternatives 90
B.5.2 TOPP Description 90
B.5.3 TOPP Setup 91
B.5.4 TOPP Evaluation 92
B.5.5 Considerations 93
B.6 Comparison of BTC and TOPP 93
B.6.1 Measurement Setup 94
B.6.2 Results Evaluation 94
B.7 Discussion and Conclusion 95
References 96

C Verification of 3G and 4G Received Power Measurements in a Crowdsourcing Android App 99
C.1 Introduction 101
C.2 The NetMap Application 102
C.3 Methodology 104
C.3.1 Tools and Scenarios 105
C.3.2 Data Processing and Evaluation 106
C.4 Results 110

C.5 Discussion 111
C.6 Conclusion 114
References 115

D On the Benefits and Challenges of Crowd-Sourced Network Performance Measurements for IoT Scenarios 117
D.1 Introduction 119
D.2 Crowd-Sourced Network Performance Measurements 121
D.2.1 Measurement collection software and metrics 121
D.2.2 Measurement setting 122
D.2.3 Measurements 122
D.3 Evaluation of Measurements 123
D.3.1 Measurement Processing Approach 123
D.3.2 Impact Evaluation Approach 125
D.3.3 Impact Evaluation 128
D.3.4 Recommended Size of R 133
D.4 Conclusion 134
References 134

E Accurate and Quality-Aware Bus Occupancy Estimation Utilizing Probabilistic Models for WLAN Probing 137
E.1 Introduction 139
E.2 Design of WiFi Probe Collection System 142
E.3 Baseline Algorithm Definition and its Performance 145
E.3.1 Definition of Baseline Algorithm 145
E.3.2 Experimental Setup 146
E.3.3 Impact of Device Presence Time Threshold 147
E.3.4 Impact of RSSI threshold 148
E.3.5 Considerations of Baseline Algorithm Results 150
E.4 Enchanced bus occupancy estimators 151
E.4.1 Estimated Number of Devices 152
E.4.2 False Positives 153
E.4.3 False Negatives 154
E.4.4 Improved Estimator 155
E.4.5 Maximum Likelihood Estimator (MLE) 156
E.5 Finding Parameters 158
E.5.1 Selection of Threshold Pairs 165
E.6 Estimated Number of People 166
E.7 Results 169
E.8 Design of Quality Indicator 173
E.9 Summary and Outlook 173
References 175

F Intelligent Parking Assistant - A Showcase of the MOBiNET Platform Functionalities 179
F.1 Introduction 181
F.2 Intelligent Parking Assistant (IPA) System Overview 182
F.3 MOBiNET platform functionalities 183
F.4 Intelligent Parking Assistant System 185
F.4.1 App Functionalities 186
F.4.2 Back-end Services and Parking Lot Information 188
F.5 Use Case 190
F.6 Evaluation of the System 191
F.6.1 Performance Indicators 191
F.6.2 Evaluation Scenario 191
F.7 Next Steps 192
References 192

Thesis Details

Thesis Title: Enhancing IoT Systems by Exploiting Opportunistically Collected Information from Communication Networks
Ph.D. Student: Lars Møller Mikkelsen
Supervisors: Assoc. Prof. Tatiana K. Madsen, Aalborg University
Prof. Hans-Peter Schwefel, Aalborg University
Assoc. Prof Rasmus L. Olsen, Aalborg University

The main body of this thesis consists of the following papers:

[A] L. M. Mikkelsen, S. R. Thomsen, M. S. Pedersen, and T. K. Madsen, "NetMap - Creating a Map of Application Layer QoS Metrics of Mobile Networks Using Crowd Sourcing." *Cham: Springer International Publishing,* 2014, pp. 544–555. [Online]. Available: `http://dx.doi.org/10.1007/978-3-319-10353-2_50`

[B] L. M. Mikkelsen, N. B. Højholt, and T. K. Madsen, "Performance Evaluation of Methods for Estimating Achievable Throughput on Cellular Connections." *Cham: Springer International Publishing,* 2015, pp. 422–435. [Online]. Available: `http://dx.doi.org/10.1007/978-3-319-23126-6_37`

[C] M. Lauridsen, I. Rodriguez, L. M. Mikkelsen, L. C. Gimenez, and P. Mogensen, "Verification of 3g and 4g received power measurements in a crowdsourcing android app," in *2016 IEEE Wireless Communications and Networking Conference,* April 2016, pp. 1–6.

[D] L. Mikkelsen, T. Madsen, and H.P. Schwefel, "On the Benefits and Challenges of Crowd-Sourced Network Performance Measurements for IoT Scenarios." submitted and under review at *International Journal on Wireless Personal Communications as a special issue of the selected papers of GWS 2016,* Dec 2016.

[E] L. Mikkelsen, T. Madsen, and H.P. Schwefel, "Accurate and Quality-Aware Bus Occupancy Estimation Utilizing Probabilistic Models for WLAN Probing." submitted and under review at Journal *IEEE Transactions on Vehicular Technology*, March 2017

[F] L. Mikkelsen, R. Toledo, and N. Agerholm, "Intelligent parking assistant - a showcase of the mobinet platform functionalities," in *2015 22nd ITS World Congress, Bordeaux, France*, Oct 2015.

In addition to the main papers, the following publications have also been made:

[1] T Schlauch, D. Beckmann, I Passchier, N. Zahariev, and L. Mikkelsen, "MOBiNET – an innovative approach for a European-wide ITS service platform," *10th ITS European Congress, Helsinki, Finland*, June 2014, TP 0187.

[2] U. Noyer, T. Schlauch, P. Cercato, and L. Mikkelsen, "MOBiNET – architecture overview of an innovative platform for European mobility services," *22nd ITS World Congress, Bordeaux, France*, Oct 2015, ITS-1980.

[3] U. Noyer, T. Schlauch, B. Wissingh, and L. Mikkelsen, "MOBiNET: Architecture and experience from a marketplace for mobility services," *11th ITS European Congress, Glasgow, Scotland*, June 2016, EU-TP0005.

[4] L. Mikkelsen, R. Buchakchiev, T. Madsen, and H. P. Schwefel, "Public transport occupancy estimation using wlan probing," in *2016 8th International Workshop on Resilient Networks Design and Modeling (RNDM)*, Sept 2016, pp. 302–308.

[5] J. Hernández-Serrano, J. Muñoz, A. Bröring, O. Esparza, L. Mikkelsen, W. Schwarzott, and O. León, "On the Road to Secure and Privacy-preserving IoT Ecosystems," *2nd International Workshop on Interoperability & Open Source Solutions for the Internet of Things (InterOSS-IoT 2016)*, Nov 2016, Stuttgart, Germany. Springer, LNCS.

This thesis has been submitted for assessment in partial fulfillment of the PhD degree. The thesis is based on the submitted or published scientific papers which are listed above. Parts of the papers are used directly or indirectly in the Introduction part of the thesis. As part of the assessment, co-author statements have been made available to the assessment committee and are also available at the Faculty. The thesis is not in its present form acceptable for open publication but only in limited and closed circulation as copyright may not be ensured.

Acknowledgments

There are many I would like to give thanks to for helping me realize this PhD. I am especially thankful to my supervisors for the guidance and support they provided in the everyday to help me keep focus and push forward. I would like to thank Tatiana K. Madsen for giving me the opportunity to write this PhD, and for tirelessly supporting me in writing papers and dealing with all aspects of my work. I would like to thank Hans-Peter Schwefel for fruitful discussions and for helping me raise the scientific level of my papers. And a thanks to Rasmus L. Olsen for supporting me in the initial phase of the PhD with experience of European project work.

I would also like to thank my colleagues and especially my office mates for support and for making the PhD work fun. I want to extend a special thanks to Dorthe Sparre for supporting me in many practical and organizational issues. During this PhD I have worked on two big European projects; MOBiNET and BIG IoT. Both projects provided me with a lot of experiences, both through good and hard times. From this I learned a lot and got the opportunity to work with professionals from all over Europe and gained a lot of friends.

I would also like to thank NEC for giving me the opportunity to stay and work in another international research environment. This was a very pleasant experience that helped me widen my horizon even more.

Finally, I would like to thank my girlfriend Nicole for believing in me and motivating me to get through this. I would also like to thank my friends and family for supporting me, especially my dear parents.

This PhD project has been a journey of discovery for me, both personally and professionally. During this project, I have experienced ups and downs, frustration and joy, confusion and certainty, all of which I am thankful as I achieve my goal.

MOBiNET project This work is partly supported through the MOBiNET project which is co-funded by the European Commission under the 7th RTD Framework Programme, Directorate General for Communications Networks, Content & Technology (FP7-ICT-2011-6.7) – Grant agreement no.: 318485.

BIG IoT project This work is partly supported through the BIG IoT project (Bridging the Interoperability Gap of the Internet of Things) funded by the European Union's Horizon 2020 research and innovation programme under grant agreement no.: 688038.

Lars Møller Mikkelsen
Aalborg University, March 31, 2017

Part I

Introduction

Chapter 1

Motivation and Problem Description

This chapter will introduce the content and problem area of this thesis by giving an overview of the background. This will serve as a motivation for working with this area. Then the specific problem area will be outlined followed by a problem statement.

1.1 Thesis Context

In the past almost 10 years there have happened an explosion like increase in number of smart devices, enabling users to be connected all the time and everywhere. This is based on the big development and improvement in the capabilities of electronic components, making them smaller, faster and cheaper. The communication technologies and networks have also improved considerably, effectively making the term "no connection" a rare exception rather than the rule. With this technology becoming available for regular users, they have quickly adapted to the new opportunities that the technology brings. The users have higher expectations to response time of devices and of the networks, but also higher expectations to availability and capabilities of services. For most parts this is supported by increased computational power of devices and increased speed and coverage of networks. Finally, in more recent years the concept of Internet of Things (IoT) has kicked off supported by the technology readiness. IoT is the concept of adding connectivity to all the things in our daily life and letting them talk to each other. This is often done in the domains of smart city, production, home automation, and other, where IoT systems consists of sensors, things, service, and applications. The goal of IoT systems is to automate certain tasks and utilizing available information of the context of the task. The increased connectivity of things is used to both provide and consume an endless stream of information generated by devices and users. When IoT systems get more relevant information to base their decisions on they become smarter. And by making them smarter they intelligently can help make all of our lives easier.

1.2 Background and Motivation

Today smart IoT systems operate as cyber physical systems, meaning that the system both has a digital and a physical context. The physical context is defined in terms of location, movement speed, number of other users, connection technology, signal strength and other. The digital context is defined in terms of user identity, services, network performance, and other. We define smart as the ability to identify relevant context information and utilize it in decision algorithms. However the context information available to an IoT system is often limited to a specific area. By giving systems access to more general context information types, both in terms of the physical context and the digital context, they are able to perform smarter decisions and deliver more relevant information to the user.

Information Types One challenge in making services smart is about what types of information to include in services. The information types will naturally depend on the category of system and the domain it operates in. For instance a parking system might need information about parking lots and their properties, road traffic, street layout, and other. While a home automation system might need information about user schedules and presence, indoor and outdoor temperature, and other. In this work we focus on a few but diverse information types that are relevant to various services operating in various domains and systems, namely network performance and people count.

The first diverse information type we will focus on is information about network performance, and specifically what performance devices can expect from the network when using it for communicating with other networked entities. IoT systems often operate connected to the Internet, why information about network performance is important to most systems. As IoT systems often are highly distributed, they often employ wireless technologies to ensure connectivity between system components. This is both to support connectivity over great geographical distances, to support high mobility of things, and to reduce connection costs. The trade-off of using wireless technologies is the network performance that inevitably will vary depending on various influencing factors, which means that the wireless connection will have a great impact on the system performance. For instance in a non-live on demand video streaming system, there is a need to periodically download the next part of the currently watched video. The system does not need to constantly download, or stream, the video. This means that the system can schedule the video download, but if it does not have information about network performance in the context of the device, it might end up accidentally scheduling download in poor connection areas. For this reason, information about the wireless network performance is of great interest.

The other diverse information type we will focus on, which also is relevant for systems operating in the public domain, is the number of users or people in the area of the system. IoT systems operate in the physical world interacting with users, or give information about physical systems that interact with users. For this reason knowing the amount of people in the area of the physical service means that we are able to say something about load and availability of the service. Again the parking system example can be used; knowing about the number of people in the area of a parking lot might be relevant in predicting parking availability.

Collection of Information Types To make the services truly smart we must be sure to have up to date information available to the system. So the challenge is how to collect the information and how to do so efficiently such that up to date information is available to the services that need it.

One approach could be to make dedicated data collection systems, where hardware is deployed to collect data in the needed locations. In this approach we can develop, adjust and place the hardware to measure exactly the type of information needed, but it would however require a lot of resources.

An alternative approach is to opportunistically collect information by exploiting hardware and technologies already present to extract the needed information. This will greatly reduce costs of hardware and deployment, but at the cost of accuracy in the obtained information. One example of already present hardware is consumer smartphones that in recent years have exploded in numbers and capabilities. This can be exploited via crowd sourcing, that is getting users to voluntarily perform small tasks to help us achieve our goal. Another example of exploiting already existing technology is by passively collecting information from deployed communication systems. This is a very common approach applied in many different domains and situations, e.g. in collecting information about presence of devices without explicitly interacting with them. In this work we will look at both crowd sourcing and passive data collection from deployed communication systems, to obtain needed information.

So it is possible to opportunistically collect the needed information while keeping the costs low, but it comes at another cost, namely the inaccuracy and uncertainty from collecting information using these approaches. When using equipment that is not solely used for the measurements it might influence the measurements. The smartphone is a good example of this issue, because it is basically acting as the Swiss army knife of electronics. It is not just a phone but also a small computer, supporting most standard PC functionalities. This is what makes the smartphone successful and interesting, but also what makes it very difficult to predict in terms of load and performance. Furthermore, the network connection is carrying all kinds of traffic generated by installed applications, making it highly fluctuating in load and performance.

Information Quality When having collected the information the challenge is how to evaluate the quality of information. This is important to know to be able to trust systems and their decisions made based on the information. Information quality can be determined based on different parameters such as freshness, accuracy or confidence, depending on the use case.

Data quality can be evaluated in terms of data freshness, i.e. how much time passed since the data was measured, logged or generated, and until the data is used. Assuming that the data is pushed from the source to destination as soon as it is ready, the main factors influencing the freshness is network delay. From an application point of view it is possible to estimate the expected freshness of received data based on knowledge of the network performance and delay. This could be estimated from the network performance information described earlier.

Another metric of data quality is accuracy, i.e. how accurate the information can be expected to be, based on the method that is used to collect the information. For instance, in the crowd sourcing case where consumer smartphones are used, the estimation of network delay or throughput might not be as accurate as when obtained using a dedicated measurement equipment. Accuracy could be determined in terms of confidence of the estimation method. This could be the case in scenarios where the information is not directly measured but instead is estimated based on other metrics. So given that probabilistic models can be derived for the characterization of estimators for certain parameters from sensor data, confidence intervals from these probabilistic models can be used as one type of quality indicators

Adding an indication of the data quality allows services and applications using the data to adapt to the quality of data. This adaption could for instance be selecting another data source, or to pass the information quality to the user giving him the choice of not trusting the system decision because the data quality is too low. Conversely if the quality is high the system could be allowed to make decisions fully autonomously.

1.3 Problem Area Delimitation

With the main challenges outlined in the previous section we will now delimit the problem areas that this thesis will concern by outlining the specific problem areas.

We want to study if we can enhance IoT systems by exploiting various information collected from communication networks. We want to collect the information in an opportunistic manner either via active measurements or via passive detection of communication networks parameters.

Figure 1.1 shows an overview of components, actors and technologies related to the problem areas that we will focus on in this thesis. The problem areas are highlighted in green, red and blue circles.

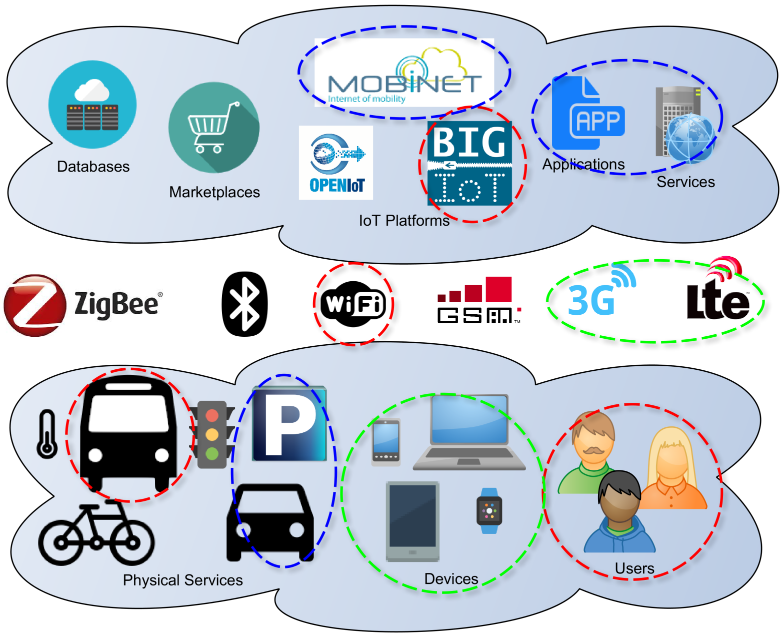


Fig. 1.1: Context of the problem areas of this thesis.

The first problem area (marked in green in Figure 1.1) we will focus on is concerning the performance of the connection between user devices and services, i.e. how to estimate the network performance, how to collect the information, and how to process it. Specifically we will focus on measurements on 3G and 4G cellular connections and evaluate how we can do this at a low cost by applying crowd sourcing, while not sacrificing accuracy and reliability of the information.

The second problem area (marked in red in Figure 1.1) we will focus on is related to the context of the user, namely the context in terms of other users, i.e. the number of people in the area. Again here we will investigate how to obtain this information in a cost effective way by deriving the number of people based on passive collection of WiFi probes. This work will address how to derive the number of people in an area from other parameters, and how to quantify the quality of the information for use by services and applications.

The third problem area (marked in blue in Figure 1.1) we will focus on is related to the functional smart IoT system that is comprised of an application, services, and data sources such as sensors and things. We will look at the specific use case of smart parking, i.e. how to make parking easier for the users. This covers how to discover services and provide the user with information about live parking availability, how to guide the user to parking lots, how to automate payment for parking, and other aspects. In doing this we will consider how to ensure high flexibility and interoperability between services, applications, platforms, etc.. We also propose how to include the previously described information types, network performance information and information about number of people, to make the system smarter.

To sum up, in this work we will focus on how to obtain the information types network performance information and number of people in an area. We will do this by opportunistically extracting information from communication networks. We will evaluate how to use these information types in connection with service discovery and service selection in the context of a distributed smart IoT system.

1.4 Problem Statement

With the problem areas narrowed down in the previous section, we now define a problem statement that we will answer in this thesis.

How can IoT systems be enhanced by utilizing opportunistically collected information about network performance and information about number of people in a limited geographic area?

In answering this question the following subquestions will be explored:

- How to obtain network performance information via crowd sourcing in an efficient and valid manner?

  In doing this we will evaluate different network performance metrics and measurement methods. Furthermore, we will explore how to collect the measurements via crowd sourcing and the impact of doing this.

- How to cost effectively estimate number of people in a limited geographic area based on passively observed network parameters and how to evaluate the quality of the estimate?

We will delimit this to be done based on observed metrics extracted from WiFi communication. Furthermore, we will do this in the scenario of estimating bus occupancy in connection with the BIG IoT project [1].

- How to create a smart IoT system that can take advantage of the collected information to perform service discovery and selection?

  We will explore this in the domain of Intelligent Transportation Systems and specifically related to smart parking. We will do this in context of the MOBiNET project [2] by creating a demonstration use case.

1.5 Outline and Overview of Contributions

The remainder of Part I will first present the state of the art in Chapter 2 which will cover the main topics related to the contributions of this thesis.

Chapter 3 covers our contributions related to network performance measurements collected using crowd sourcing. First we present NetMap [3], a crowd sourcing system that we created for performing and collecting network performance measurements from smartphones. Next we evaluate methods for collecting network performance metrics, specifically achievable throughput and received power. Finally we address one of the challenges arising from crowd sourcing collected measurements, namely areas with sparse measurements. This chapter is based on Paper A, Paper B, Paper C and Paper D.

Chapter 4 covers our contribution to people counting based on passively collected WiFi probes. First we present the WiFi probe collection system we created for this purpose. Next we address how we estimate the number of people on a bus based on WiFi probes collected with the system, and how we improve the estimator using probabilistic modeling. Finally we create a quality indicator of the obtained estimate that can be used by applications as users of the people count information. This contribution is done in connection with the BIG IoT project [1]. The contribution described in this thesis will be used as part of several use cases related to public transport. This chapter is based on Paper E

Chapter 5 covers our contribution to enhancing IoT systems by including contextual information. First we present an intelligent parking assistant system that we created. The system uses the MOBiNET platform for discovering services and parking information sources relevant to the current location. The system supports the user with live parking availability information and automatically initiates and pays for parking sessions, based on user location and movement. Next we present how the previously presented information types, network performance and people count, can be used to enhance service discovery and selection, and how it can be exploited by a system such as the intelligent parking assistant system. This contribution is done in connection with the MOBiNET project [2]. We developed the parking system to demonstrate the capabilities of the MOBiNET platform. This chapter is based on Paper F.

Chapter 6 we conclude on the contributions and findings, and present an outlook on future work.

Part II contains the papers that this thesis is based on.

References

[1] BIG IoT, "Big iot project," March 2017. [Online]. Available: http://big-iot.eu/

[2] MOBiNET, "Mobinet project," March 2017. [Online]. Available: http://mobinet.eu/

[3] NetMap, "Netmap," March 2017. [Online]. Available: http://netmap.aau.dk/

Chapter 2

Related Work

In this chapter we will present the state of art related to the main topics covered by the contributions of this thesis. The main topics are end-to-end network performance estimation, crowd sourcing of network performance, people counting, and service discovery in IoT platforms.

2.1 End-to-End Network Performance Estimation

In this work we will estimate the end-to-end network performance, in our case meaning the performance of the connection measured on the transport layer in the TCP/IP model. Furthermore, we will evaluate the performance between a device and a server. In our case the connection will include a wireless cellular connection, the internal service provider network, Internet exchange, and the danish research network. For the wireless cellular connection we will mainly focus on 3G and 4G, as these are the most common networks used by consumer smartphones. The internal service provider network covers the data traffic path from the cellular base station (eNodeB or Node B) to the edge of the network where traffic is exchanged with other networks, i.e. the Internet exchange. The danish research network is included because this is where we have placed our measurement server.

2.1.1 Network Performance Map (NPM)

We include the cellular connection in our end-to-end connection because we want to create a geographical network performance map (NPM) based on actual network performance measurements. A NPM is showing the expected performance that applications can expect at a certain location in terms of metrics such as throughput and latency. The application is in our case running on a smartphone and communicating with a back end server. The NPM can also show the performance variations over time.

Today the most common versions of NPMs are those provided by mobile network operators (MNOs), where users can select a location and see the expected network performance in terms of data speed [43] [41] [42]. These are based on coverage maps, i.e. showing the coverage of the cellular signals based on propagation models and a limited number of signal strength measurements. Based on this the expected data throughput is calculated and presented on the map. It is however not specified what is specifically meant by throughput, but only that it is calculated different than the rates that are advertised (in subscriptions), and that the calculations are theoretical. Furthermore, they also indicate that nature, buildings, distance to base station, and number of concurrent users at a base station will influence the experienced throughput.

Another NPM is provided by OpenSignal [27], that employs a crowd sourcing to collect data. Also here the main focus is on making a coverage map of signal strength, but they also collect throughput and latency measurements. The advantage here is that the NPM is based on actual measurements, and not calculated performance based on signal strength coverage. The OpenSignal NPM is showing more sporadic coverage that the maps of the MNOs, because the measurement coverage is more sporadic.

2.1.2 Network Performance Metrics

As mentioned earlier the NPM should present the performance that applications can expect. For this reason we will focus on two main metrics; throughput and delay.

Throughput is a metric defining how much data can be transferred on a connection per time interval. This is also referred to as achievable throughput [6] as it describes the throughput rate that can be achieved in the given setting where it is measured. This can be seen as a contrast to capacity [5], or bandwidth, estimation, where the maximum capacity of a connection is the absolute upper bound for achievable throughput, however often it is lower due to other traffic and protocol overhead. In our case we define achievable throughput as the amount of payload data that can be transferred on the transport layer. The throughput metric is highly relevant to many applications, e.g. streaming services and file download, as the achievable throughput determines the speed at which the communication can be done.

Delay is a metric relating to the time it takes to send a single packet over a connection. This relates to the metrics one-way delay, round-trip time, and jitter. One-way delay [2] refers to the time it takes to send a packet between two end-points, e.g. a device and a server. This can be difficult to measure due to the need for clock synchronization between the two entities. Round-trip time (RTT) [1] refers to the time it takes to send a packet back and forth between two end-points, e.g. from a device to a server and back to the device, excluding the processing time at the server. Jitter [9] refers to the variation of the delay metric, which could be one-way delay or RTT. The delay metrics are relevant as they impact several aspects of communication. For instance RTT and jitter have a great impact on the performance of the TCP protocol, due to how sends control packets (ACK and SYN) back and forth between the communicating entities. Another example the delay impact on as system that polls for updates in the background, or a systems where many small packets are sent back and forth, e.g such as when loading websites.

2.1.3 Measurement Methods

As described earlier we are interested in the network performance measured on the transport layer in the TCP/IP model, and particular the performance of the protocols TCP and UDP as these are the most common. The TCP protocol employs congestion control and acknowledgments to ensure reliable communication and congestion avoidance, while UDP does not. This means that the network performance metrics will influence the two protocols differently. For this reason we will measure achievable throughput and RTT using both protocols.

Achievable throughput is a complex metric because several factors are influencing it. The choice of protocol to measure with will influence the result in terms of different overhead. Here specifically the difference between TCP and UDP can be mentioned. Furthermore, the available bandwidth will play a factor as the upper limit of the achievable throughput. And if TCP is used the RTT will also have an impact on the measured results.

Available bandwidth is often estimated using methods that can be divided into two categories; Probe Gap Models (PGM) and Probe Rate Models (PRM) [24]. PGM methods estimate the available bandwidth using a train of probe packets transmitted with a fixed rate. This rate is chosen based on knowledge of the tight, or bottleneck, link on the measured path [11] [15] [8]. PRM methods estimate the available bandwidth using several trains of probe packets where the rates of the trains are varied, in order to search for the rate that match the available bandwidth of the link [32] [30] [12] [11].

Achievable throughput is often measured using brute force methods, such as Bulk Transfer Capacity (BTC) [31], where as much data as possible is transmitted over a period of time. The throughput is then calculated as the ratio of data over time. When using TCP in BTC, the TCP congestion control avoids congestion on the connection, by throttling the transmission rate. Furthermore, TCP will also send out a lot of control packets on the connection in addition to the measurement traffic. When UDP is used in the same manner as TCP in BTC, the connection could possibly be flooded as there is no congestion control. This means that there are more requirements to rate control of the sender when using UDP, but also more options to control the transmission. In [28] the authors propose a method using UDP that first estimates available bandwidth, and then achievable throughput by exploiting the control of transmission of UDP packets.

When estimating end-to-end RTT on the transport layer typically either TCP or UDP is used. This can be done by sending a payload packet from a client to a server, and the server then replies as fast as possible. When using TCP the connection should first be set up before transmitting the actual measurement traffic. Alternatively the control packets of TCP, such as TCP SYN and TCP RST [16], can be exploited to measure the RTT. When using UDP measuring RTT is more straight forward as no connection is established but the packet is just sent. This approach is applied in [38] where 20 RTT samples are done.

2.2 Crowd Sourcing Network Performance Measurements Using Smartphones

Users use consumer devices such as smartphones for many tasks, such as web browsing, sending and receiving email, chatting, calling, streaming, gaming, and so on. Any of these tasks are done while being mobile and moving around. However, the devices are rarely used 100% of the time, leaving them idle and unused for time durations. This can be exploited by assigning small tasks to the devices to carry out that will serve a greater purpose, which is how we define crowd sourcing. The mobility of the devices and their communication capabilities make smartphones suitable to be used for estimating network performance measurements for use in creating a NPM.

There exists several solutions for performing crowd sourced network performance measurements. These can mainly be divided into two groups; app based and web based. In the web based category the user have to visit a webpage to perform a measurement. Examples in this category are SpeedTest [25] and SpeedOf.me [39]. Both of these mainly focus on measuring the throughput speed of the connection. These systems are focused on personalized measurements, why no NPM is presented. However, they do present statistics about measurements obtained for various operators.

In the app based category the user installs a client application on his device, from which the various measurement methods are executed and context information is collected. Examples of app based systems are MIST [46], SamKnows [34], MobiPerf [19], NetRadar [37], and OpenSignal [27]. These systems perform various combinations of estimating achievable throughput, available bandwidth, RTT and various connection capability testing. The systems mostly require user interaction to perform the measurements, but some of them perform automated signal strength logging.

MobiPerf [19] provides a NPM showing individual measurements, but this system is more focused on personal measurements, i.e. not contributing to a general NPM. NetRadar [37] provides a map, but at the time of writing this no data is shown. OpenSignal [27] provides a general NPM based on the measurements performed. However, due to the main focus being on signal strength, many locations does not contain information about latency or throughput, but only signal strength. These approaches show a common problem of creating a NPM, namely the general lack of measurements to properly cover areas in the map. This is a contrast of the potential of crowd sourced measurements and the possibility to achieve great geographical measurement coverage as all consumer devices are potential measurement devices. This could be solved by increasing the measurement frequency of the systems.

In creating a NPM the goal is to be able to show a close to continuous coverage. For this reason the measurements need to be aggregated within smaller areas. In [17] they propose an approach for measurement aggregation, where measurements from different devices should be assigned different weights. In their proposed approach they obtain these weights via an optimization function between aggregated result and proposed weights.

An advantage of crowd sourcing network performance measurements is the low cost of measurement collection, when compared to performing measurement campaigns using dedicated measurement equipment. But this comes as a trade-off of measuring with non-dedicated measurement equipment, such as smartphones. This means that each measurement is subject to a number of influencing factors such as mobility of the device, wireless signal interferences, signal path obstructions, cross traffic on commercial networks, other apps putting load on the device, and other. This is demonstrated in [16] where the accuracy of RTT measurements are shown to be influenced by the virtual machine that Android apps are executed in, but also by delays caused by the hardware interaction (drivers and kernel).

The variability of measurements caused by the network and location is interesting as it says something about the reliability of the network. For this reason the performance variability information should be kept in the NPM. This is however not the case in any of the presented NPMs.

2.3 People Counting Techniques

People counting is the field of estimating the number of people in a confined geographical area, such as on a street segment, in part of a concert venue, or on a bus. The information about number of people can be used by services to inform users about availability of physical services, or to help understand user load. To automate people counting different technologies are utilized, which yield different advantages and disadvantages.

Generally people counting techniques can be divided into two groups; device free or device based approaches. A commonly described device free approach is video based [29] [40] [13] [35]. In this approach a video feed, e.g. from a surveillance camera, is used where different types of image processing are done to obtain people count information or people flow information. The video based approach is accurate in terms of it counting what it sees, but it is limited by only covering line of sight from the camera.

Another device free approach is radio frequency profiling which is done using technologies such as cellular [44] or WiFi [14] [33]. In radio frequency profiling the signal response of a room or small area is evaluated. The system is then trained with different situations where the number of people and their location in the evaluated area is known. The system then provides an estimated number of people given the current radio frequency profile. The radio frequency profiling approach is less limited in coverage, but requires significant training for the given situation that it is deployed in.

Device based approaches can again be split into two; passive or active. In active device based approaches there is communication between the people counting system and a device carried by the person being counted. The communication could be to an app installed on the user smartphones, or to a Bluetooth tag [3]. This is potentially very accurate but it does require people volunteering to participate.

In passive device based approaches a system is designed to passively collect signals emitted by devices for other purposes. This has been done by passively collecting Bluetooth probes [22] [20] [45] to estimate number of people visiting malls, or to estimate crowd density at football stadiums. A similar approach has been done using WiFi probes [10] [21] to estimate the number of people on public transport or to track people movement. The advantage is that today the penetration of Bluetooth and WiFi technologies is high, making the potential coverage good. The disadvantage of the passive device based approaches is that they require some processing after collecting the signals to obtain the estimated number of people.

In this work we will perform bus occupancy estimation by applying a passive device based approach similar to the approach presented in [10]. However, our approach will differ in how we process the collected WiFi probes, which will be based on probabilistic modeling.

2.4 IoT Platforms and Service Discovery

In recent years several IoT platforms have emerged that offer various functionalities to service and IoT system developers. An IoT platform is a combination of software components offering standard functionalities, such as management of services, analytics of usage, and service discovery. Examples of IoT platforms are OpenIoT platform [26], Kaa IoT Platform [7], Yucca Platform [36], and MOBiNET Platform [18]. In the following we will focus on service discovery and the characteristics of this.

With services we mean web services that provide APIs for interaction with other services and applications. Services can provide information or functionalities and applications are front end software that users interact directly with, meaning that they have a graphical interface.

Service discovery is the operation of finding relevant services to interact with. In [23] the authors dissect the concept of web service discovery by dividing it into five characteristics; storage of service descriptions, formalization of service characteristics, matching between requests and service descriptions, automatic or manual service selection, and characteristics to perform service selection based on.

To enable discovery of services the services must be described which can be done using a syntax description language or a semantic description language. Syntactical approaches typically focus on describing the service interface in terms of parameters and operations offered by the service. Semantical approaches focuses on describing the service interface in terms of ontologies and vocabularies to allow for machine interpretation of descriptions.

The storage of service descriptions can be done centrally or distributed. The central approach is typically realized using a directory or registry, where service descriptions are added. The distributed approach there is no central entity storing all service descriptions, but rather several entities. This could for instance be realized using peer-to-peer systems.

To perform the discovery of services a matchmaking must be done based on input from the searching entity. The matchmaking is dependent on the applied service description approach. For syntactical description approaches the matchmaking is done based on keyword or category based searches. For semantical description approaches the matchmaking is done based on ontologies. Some systems also employ a context based service discovery, where the context of the searching entity is utilized in finding relevant services.

The discovery can be done manually or automatically where in the manual approach developers manually discover services during development and utilize the result. Automatic service discovery can be done by the searching entity being programmed to perform the discovery, or by the service discovery system keeping the discovery query and returning new results when they occur.

Finally there must be done a selection among the discovered services, which is done based on functional or non-functional characteristics of the services. Functional characteristics is what the service do, e.g. in terms of input and output parameters. Non-functional characteristics relate to QoS in terms of availability, privacy, quality of output, and other.

As an example the MOBiNET platform offers service discovery by employing a centralized service directory where service descriptions are stored. Services are described syntactically and the matchmaking is done as a hybrid between keyword, context and category based. This is realized in MOBiNET Service Directory by offering a range of filtering options. MOBiNET offers both manual and automatic service discovery, supporting both developers during design and services during runtime. The service selection is done as a hybrid between functional and non-functional characteristics of services. MOBiNET differs from other platforms in that it focuses mainly on the mobility domain. Furthermore, it also focuses on the interactions between businesses in terms of identification and billing.

We will in this work create a smart system supporting users in finding and paying for parking. The system will demonstrate the advantages of utilizing automated context based service discovery, specifically provided by the MOBiNET platform. We will propose how to improve on context based service discovery selection, in not just looking at location but also include network performance information and information about number of people.

With many different IoT platforms available there is a wide range of APIs available for developers to choose from when developing their systems. So if a system should be interoperable with several platforms the needed implementation efforts scale accordingly. This is the problem that the BIG IoT project [4] attempts to solve, by developing a common API for platforms to interface with. Service and application developers now only need to interface with one API to gain access to services and data from any platform that implements the BIG IoT API. The BIG IoT API is currently under development, but eight IoT platforms have committed to implement the API.

References

[1] G. Almes, S. Kalidindi, and M. Zekauskas, “A round-trip delay metric for ippm,” Internet Requests for Comments, RFC Editor, RFC 2681, September 1999.

[2] G. Almes, S. Kalidindi, M. Zekauskas, and A. Morton, “A one-way delay metric for ip performance metrics (ippm),” Internet Requests for Comments, RFC Editor, STD 81, January 2016.

[3] A. Basalamah, “Sensing the crowds using bluetooth low energy tags,” *IEEE Access*, vol. 4, pp. 4225–4233, 2016.

[4] BIG IoT, “Big iot project,” March 2017. [Online]. Available: http://big-iot.eu/

[5] P. Chimento and J. Ishac, “Defining network capacity,” Internet Requests for Comments, RFC Editor, RFC 5136, February 2008.

[6] B. Constantine, G. Forget, R. Geib, and R. Schrage, “Framework for tcp throughput testing,” Internet Requests for Comments, RFC Editor, RFC 6349,

August 2011, http://www.rfc-editor.org/rfc/rfc6349.txt. [Online]. Available: http://www.rfc-editor.org/rfc/rfc6349.txt

[7] CyberVision Inc, "Kaa iot platform," March 2017. [Online]. Available: https://www.kaaproject.org/

[8] A. Delphinanto, T. Koonen, and F. den Hartog, "End-to-end available bandwidth probing in heterogeneous ip home networks," in *2011 IEEE Consumer Communications and Networking Conference (CCNC)*, Jan 2011, pp. 431–435.

[9] C. Demichelis and P. Chimento, "Ip packet delay variation metric for ip performance metrics (ippm)," Internet Requests for Comments, RFC Editor, RFC 3393, November 2002, http://www.rfc-editor.org/rfc/rfc3393.txt. [Online]. Available: http://www.rfc-editor.org/rfc/rfc3393.txt

[10] M. Handte, M. U. Iqbal, S. Wagner, W. Apolinarski, P. J. Marrón, E. M. M. Navarro, S. Martinez, and S. I. Barthelemy, "Crowd density estimation for public transport vehicles," *CEUR workshop proceedings*, no. 1133, pp. 315–322, 2014.

[11] N. Hu and P. Steenkiste, "Evaluation and characterization of available bandwidth probing techniques," *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 6, pp. 879–894, Aug 2003.

[12] A. Johnsson, M. Bjorkman, and B. Melander, "An analysis of active end-to-end bandwidth measurements in wireless networks," in *2006 4th IEEE/IFIP Workshop on End-to-End Monitoring Techniques and Services*, April 2006, pp. 74–81.

[13] P. Karpagavalli and A. V. Ramprasad, "Estimating the density of the people and counting the number of people in a crowd environment for human safety," in *2013 International Conference on Communication and Signal Processing*, April 2013, pp. 663–667.

[14] A. E. Kosba, A. Saeed, and M. Youssef, "Rasid: A robust wlan device-free passive motion detection system," in *2012 IEEE International Conference on Pervasive Computing and Communications*, March 2012, pp. 180–189.

[15] M. Li, M. Claypool, and R. Kinicki, "Wbest: A bandwidth estimation tool for ieee 802.11 wireless networks," in *2008 33rd IEEE Conference on Local Computer Networks (LCN)*, Oct 2008, pp. 374–381.

[16] W. Li, R. K. P. Mok, D. Wu, and R. K. C. Chang, "On the accuracy of smartphone-based mobile network measurement," in *2015 IEEE Conference on Computer Communications (INFOCOM)*, April 2015, pp. 370–378.

[17] Y. Li, J. Gao, P. P. C. Lee, L. Su, C. He, C. He, F. Yang, and W. Fan, "A weighted crowdsourcing approach for network quality measurement in cellular data networks," *IEEE Transactions on Mobile Computing*, vol. 16, no. 2, pp. 300–313, Feb 2017.

[18] MOBiNET, "Mobinet - internet of mobility," March 2017. [Online]. Available: http://mobinet.eu/

[19] MobiPerf, "Mobiperf," March 2017. [Online]. Available: https://sites.google.com/site/mobiperfdev/

[20] A. Morrison, M. Bell, and M. Chalmers, "Visualisation of spectator activity at stadium events," in *2009 13th International Conference Information Visualisation*, July 2009, pp. 219–226.

[21] A. B. M. Musa and J. Eriksson, "Tracking unmodified smartphones using wi-fi monitors," in *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems*, ser. SenSys '12. New York, NY, USA: ACM, 2012, pp. 281–294. [Online]. Available: http://doi.acm.org/10.1145/2426656.2426685

[22] D. Namiot and M. Sneps-Sneppe, "On the analysis of statistics of mobile visitors," *Automatic Control and Computer Sciences*, vol. 48, no. 3, pp. 150–158, 2014. [Online]. Available: http://dx.doi.org/10.3103/S0146411614030043

[23] W. J. Obidallah, U. Ruhi, and B. Raahemi, "Current landscape of web service discovery: A typology based on five characteristics," in *2016 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*, Oct 2016, pp. 678–683.

[24] O. Olvera-Irigoyen, A. Kortebi, L. Toutain, and D. Ros, "Available bandwidth probing in hybrid home networks," in *2011 18th IEEE Workshop on Local Metropolitan Area Networks (LANMAN)*, Oct 2011, pp. 1–7.

[25] OOKLA, "Speedtest," March 2017. [Online]. Available: http://www.speedtest.net/

[26] OpenIoT, "Openiot - the open source internet of things," March 2017. [Online]. Available: https://github.com/OpenIotOrg/openiot

[27] OpenSignal, "Opensignal," March 2017. [Online]. Available: https://opensignal.com/

[28] T. Oshiba and K. Nakajima, "Quick and simultaneous estimation of available bandwidth and effective udp throughput for real-time communication," in *2011 IEEE Symposium on Computers and Communications (ISCC)*, June 2011, pp. 1123–1130.

[29] M. Parzych, A. Chmielewska, T. Marciniak, and A. Dąbrowski, "New approach to people density estimation based on video surveillance," in *2014 Signal Processing: Algorithms, Architectures, Arrangements, and Applications (SPA)*, Sept 2014, pp. 94–99.

[30] A. K. Paul, A. Tachibana, and T. Hasegawa, "An enhanced available bandwidth estimation technique for an end-to-end network path," *IEEE Transactions on Network and Service Management*, vol. 13, no. 4, pp. 768–781, Dec 2016.

[31] R. Prasad, C. Dovrolis, M. Murray, and K. Claffy, "Bandwidth estimation: metrics, measurement techniques, and tools," *IEEE Network*, vol. 17, no. 6, pp. 27–35, Nov 2003.

[32] V. J. Ribeiro, R. H. Riedi, R. G. Baraniuk, J. Navratil, and L. Cottrell, "pathchirp: Efficient available bandwidth estimation for network paths," in *Passive and active measurement workshop*, 2003.

[33] A. Saeed, A. E. Kosba, and M. Youssef, "Ichnaea: A low-overhead robust wlan device-free passive localization system," *IEEE Journal of Selected Topics in Signal Processing*, vol. 8, no. 1, pp. 5–15, Feb 2014.

[34] SamKnows, "Samknows," March 2017. [Online]. Available: https://www.samknows.com/

[35] M. Shen, T. Sun, X. Jiang, and K. Xu, "Crowd counting estimation in video surveillance based on linear regression function," in *2016 9th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, Oct 2016, pp. 60–65.

[36] Smartdatanet, "Yucca iot platform," March 2017. [Online]. Available: https://www.smartdatanet.it/yucca-platform.html

[37] S. Sonntag, J. Manner, and L. Schulte, "Netradar - measuring the wireless world," in *2013 11th International Symposium and Workshops on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt)*, May 2013, pp. 29–34.

[38] S. Sonntag, L. Schulte, and J. Manner, "Mobile network measurements - it's not all about signal strength," in *2013 IEEE Wireless Communications and Networking Conference (WCNC)*, April 2013, pp. 4624–4629.

[39] SpeedOf.me, "Speedof.me," March 2017. [Online]. Available: http://speedof.me/

[40] Y. Tabuchi, T. Takahashi, D. Deguchi, I. Ide, H. Murase, T. Kurozumi, and K. Kashino, "Spatial people density estimation from multiple viewpoints by memory based regression," in *2014 22nd International Conference on Pattern Recognition*, Aug 2014, pp. 2209–2214.

[41] TDC, "Tdc dækningskort," March 2017. [Online]. Available: http://daekning.tdc.dk/tdcnetmap_ext_tile/Default/mobile

[42] Telenor DK, "Telenor dækningskort," March 2017. [Online]. Available: https://www.telenor.dk/kundeservice/drift-og-dakning/dakning/dakningskort

[43] Telia DK, "Telia dækningskort," March 2017. [Online]. Available: http://daekning.telia.dk/TelNetMap_Main_Tile/Default/mobile

[44] Z. Tian, L. Shao, M. Zhou, and X. Wang, "A highly-accurate device-free passive motion detection system using cellular network," in *2016 IEEE Wireless Communications and Networking Conference*, April 2016, pp. 1–6.

[45] J. Weppner and P. Lukowicz, "Bluetooth based collaborative crowd density estimation with mobile phones," in *2013 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, March 2013, pp. 193–200.

[46] M. P. Wittie, B. Stone-Gross, K. C. Almeroth, and E. M. Belding, "Mist: Cellular data network measurement for mobile applications," in *Broadband Communications, Networks and Systems, 2007. BROADNETS 2007. Fourth International Conference on*, Sept 2007, pp. 743–751.

Chapter 3

Crowd Sourcing Tool for Network Performance Measurements

This chapter will introduce the contributions related to crowd sourcing of network performance measurements. We first present NetMap, a system we have created for performing network performance measurements using crowd sourcing. With NetMap we perform and collect low cost network performance measurements. Next we evaluate methods for estimating achievable throughput with the goal of finding a method suited for crowd sourcing. We also verify received power measurements by comparing consumer smartphone measurements with professional equipment measurements. Finally we address the challenge of creating a network performance map from crowd sourced collected measurements. This chapter is based on Paper A, Paper B, Paper C and Paper D.

3.1 Problem Description and Delimitations

Today when users want to know what performance to expect from the network at a certain location, e.g. in terms of achievable throughput and delay, they are left with few options. These options are often limited to the network performance maps (NPM) provided by mobile network operators (MNO), such as presented in Section 2.1. But also systems can use NPMs, e.g. to predict the network performance at a certain location and time, based on previous measurements. Knowledge of network performance can be used for optimizing services, e.g. in optimizing large file downloads [2]. Alternatively the information can be used for intelligently scheduling when to cache data in a video streaming scenario. But the MNOs NPMs are not the result of extensive network performance measurements, but calculated based on signal strength propagation models. This means that they do not contain information about actual experienced performance from user devices. So in order to make a NPM which is based on measurements on real networks a lot of measurements must be performed.

It is not trivial to obtain the needed information to create a NPM. One challenge is how to measure the metrics that go into the creation of the map. There have been done much research on the theory of measuring network performance, but in practice it is not trivial to measure on a live mobile network due to variances and influencing factors, and due to the vast coverage area. Another challenge is the cost of collecting the information needed for the NPM, as dedicated measurement scenarios are costly and take time. Finally when the needed information have been collected to create a NPM, the information is only valid for a limited time due to network dynamics. For instance when existing base stations go off line or new are deployed, when the environment changes, or as user load shifts.

One possible solution to some of these challenges is crowd sourcing, which is the approach we have chosen in this work. We have created a tool for performing and collecting crowd sourced network performance measurements of network performance. By creating this tool we show that it is possible to perform and collect great amounts of network performance measurements in a cost effective way, while providing a sufficient accuracy of measurements. We present this tool in the following sections along with how to overcome the challenges of measurement collection, adjusting and verifying measurement methods to the crowd sourcing setting, and how to deal with heterogeneous measurement distribution.

3.2 Realization of System for Cellular Network Performance Measurements Using Crowd Sourcing

Motivated by the need for a NPM based on measurements on real networks from user devices, we have created NetMap, a system for performing and collecting network performance measurements using crowd sourcing. NetMap is presented in Paper A that describes the system and how the division of tasks in the system is done.

NetMap consists of an Android client application [11], a measurement point that serves measurements, a data collection point where results are submitted to, and a presentation point that processes and presents the NPM (preliminary NPMs can be seen here [12]). In Figure 3.1 the NetMap system components and their interaction can be seen.

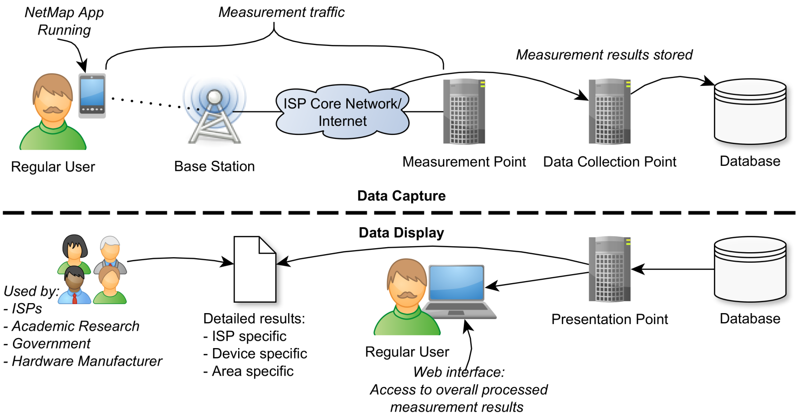


Fig. 3.1: NetMap overview (From Paper A (edited)).

The application automatically performs periodical measurements of the end-to-end connection between the device and a measurement server. When a certain number of measurements have been performed the results are submitted to a data collection point, which stores the measurements in a database. The network performance is measured in terms of application layer round trip times (RTT) using TCP and UDP, application layer achievable throughput measured using bulk transfer capacity, packet loss, connectivity indication. Also physical layer connection metrics are collected in terms of network context state switch and received signal strength indication (RSSI). Furthermore, a wide range of context information parameters are collected, such as location, movement speed, network operator, cell id, battery status, screen state, and other.

In the creation of the measurement system, and in analysis of the collected data, several challenges were considered. One challenge is how to avoid introducing bias toward measurements on one mobile network over another. For this reason we placed the measurement server on the danish research network. This means that measurement traffic goes from the mobile device via the cellular connection, through the Internet service provider network, and through an exchange point to get to the server on the research network. In that way we can compare measurements performed on different mobile networks.

A second challenge considered is that there are a number of limitations when measuring network performance using consumer devices. One being that there is no or only limited access to lower layer parameters [15]. Another is that there will be other processes running on the device during measurements requiring resources [9]. In NetMap the network performance is measured on the transport layer, i.e. the same layer as applications will use for communicating with services, which ensures the usability of the measurements. There will always be other applications running on the devices, however only a subset of them will communicate data while the measurements are taking place. This will give a true image of how the network performance that applications can expect, both in terms of the network and the connection, but also in terms of other applications using the interface.

The third challenge considered is how to interact with the user of the device. This is important as if the user is not motivated, he will leave the program and not contribute more measurements. For this reason the measurements are presented as averages in areas of a certain size. In that way location specific results of the individual user is not visible to the public. But to engage the individual user he is presented with his personal measurements in the application. This serves as motivation for the individual user to keep NetMap measuring.

With the creation of NetMap we have demonstrated how to make a system for performing cost effective network performance measurements. This is a valuable step towards creating a NPM based on actual measurements performed by the devices using the networks. The combination of collected network performance metrics and the collected context information allows for interesting analysis of the data to be performed. Paper A gives examples of the capabilities of measurements collected using NetMap when analyzing the collected data. Particularly it is shown that the collected data can show indications of network load, or cross traffic, that it can show indication of influences on the measured performance from the area where the measurements were performed, and that it can show the influence of the device model on the network performance. The results are however limited in the sense that a limited amount of measurements have been performed with the system, and for that reason the work on realizing the NPM is limited accordingly.

3.3 Estimating Network Performance Metrics in a Crowd Sourcing Setting

NetMap uses existing metrics and well defined measurement methods. However they are used in a not that well tested setting, namely in crowd sourcing and on consumer smartphones. For this reason it is important to consider how to reduce resource usage of measurement methods, particularly in relation to achievable throughput estimation. Furthermore, as the measurements are not performed with dedicated and professional measurement equipment it is important to evaluate how these measurements compare to those obtained via professional equipment. This is covered in the following.

3.3.1 Evaluation of Methods for Estimating Achievable Throughput

Achievable throughput is often used as the only metric when describing connections, e.g. when network operators sell subscriptions [16], or when network requirements are listed by streaming services [10]. For this reason it is important to estimate the achievable throughput, but this can be costly in terms of data usage as it is often done using brute force approaches where as much data as possibles transferred within a time interval. When obtaining the estimate on dedicated measurement equipment the data usage is not a problem, but on consumer smartphones resources are limited.

For this reason we have evaluated methods for estimating the achievable throughput on 3G and 4G cellular connections. This is presented in Paper B. Achievable throughput in this case refers to the throughput rate that can be achieved on the transport layer, meaning the throughput rate that applications experience. From one measurement the mean throughput rate is obtained, which is used in the NPM. The methods evaluated in the paper are Bulk Transfer Capacity (BTC) and Trains of Packet Pairs (TOPP).

Applications are often using TCP for communication with services and servers. For this reason the common method for estimating achievable throughput is used BTC that uses TCP. BTC is a straight forward method of measuring achievable throughput that transfers as much data as possible within a time interval. This means that it uses a lot of data depending on the capabilities of the device and the network, only controlled by the measurement duration. TCP is the most used transport protocol (mainly due to HTTP traffic), and for this reason the result of BTC will show quite well how most connections will perform. For this reason BTC is used as an indicator of the ground truth in the paper, meaning that results from the other method are compared the the results of BTC.

The goal is to find another method for obtaining similar results as BTC in terms of mean achievable throughput rate and variance, and doing so while using less data. The choice of TOPP is made due to it being a probe rate model as it is simple and does not have as high requirements to measurement equipment as probe gap models, in terms of timing of packets. TOPP uses UDP to send a train of packets of equal size, and when received the packet dispersion is logged. When the number of packets is sufficiently high the dispersion becomes an average packet delay, which is used in calculating the throughput rate [7].

There are a number of challenges of using TOPP as an alternative. TOPP uses UDP, which is different from BTC that uses TCP. TCP adjusts according to the connection, i.e. throttling the transfer rate. This is done based on control packets and RTT of the connection. UDP has no such throttling capabilities, but instead sends packets without any control. However, both protocols are subject to the same underlying physical channel, i.e. 3G or 4G wireless connection and the Internet service provider network. The goal is to estimate the capabilities of this connection.

Another point is that TOPP was originally developed to estimate available bandwidth bottlenecks [8] on wired connections. In this case it is used to estimate achievable throughput on wireless 3G and 4G connections. On this wireless connection the packets will be subject to cross-traffic, channel conditions, device mobility, and channel access schemes. Of these aspects only cross-traffic is present in wired connections. In [14] it is found that probe tools for estimating available bandwidth on wired connections in fact measure achievable throughput on CSMA/CA connections.

Considering these points the results obtained with TOPP and results obtained using BTC are compared in Paper B. The results are obtained from measurements performed on live 3G and 4G cellular networks. TOPP is evaluated with different settings, to find a good trade-off between low data and time usage to obtain the estimate versus obtaining an estimate similar to BTC.

The paper shows that TOPP is a good candidate for estimating achievable throughput when comparing TOPP results with results from BTC as the ground truth. This is in terms of similarity in result values and results behavior, but also in terms of TOPP being much cheaper in terms of data usage. The results are however only obtained in a limited scenario, i.e. few measurements and at two times of day. To fully validate this approach more extensive measurements should be performed. Furthermore, the difference between 3G and 4G connections should also be explored further, as the two technologies might impact the results differently.

3.3.2 Evaluation of Received Power Measurements Obtained from Smartphones

Received power is a parameter that indicates how strong the connection is between the smartphone and the base station. Poor received power will negatively influence the maximum performance of the connection that can be achieved between device and base station. For this reason it is of interest to MNOs for fixing "dead" spots in the network. For researchers data about received power is relevant to study network problems with the goal of coming up with solutions. And finally it is relevant to users in helping them in selecting the MNO that provides the best service where it is relevant to the users.

Received power measurements are typically done using professional measurement equipment by MNOs. But it can be expensive and time consuming collecting measurements, even in small dedicated measurement campaigns in limited areas. Here crowd sourcing has a clear advantage when it comes to measurement quantity and coverage. So now the question is if received power measurements performed from consumer smartphones can be used as a replacement of measurements from professional measurement equipment.

In Paper C we have performed such verification, and we show that the measurements are sufficiently accurate to track shadow fading and path loss. This is done by comparing measurements performed using the NetMap application installed on regular smartphones, with measurements performed on professional measurement phones and measurements performed using a radio network scanner. This means that received power measurements collected with NetMap can be used for creating a coverage map, as it can show the signal variations.

We compare measurements performed in four different scenarios, covering both urban and highway measurements, and both walking and driving while performing measurements. The measurements were compared using a power offset between the smartphone and the scanner, to account for the lower antenna gain of the smartphones.

The results in Paper C verifies that low cost tools such as NetMap can be used for obtaining received power measurements. It also showed the trade off there is in terms of measurement value resolution, i.e. often received power values are only updated every second, and in steps of 2 db. The results are limited by being based on a few devices, meaning that it is unknown if other device models would yield the same results. Furthermore, currently the received power is only loosely coupled with RTT measurements, making it difficult to say something about the performance of RTT in terms of received power.

3.4 Dealing with Heterogeneous Measurement Distribution

When creating a NPM the geographical area is divided into a grid, where each tile covers a limited area. Within each tile the measurements are aggregated in a mean value, e.g. mean achievable throughput and mean RTT. But also information about variance of the measurements in the tile is included in the map. The measurements within a tile may vary due to e.g. if the user holds the device in his hand or in his pocket, if there are many or few other users connected to the same base station, if there is little or much cross traffic on in the distribution network, and other. Some tiles will however have too few measurements to calculate a mean value with sufficient confidence. This is caused by users not going all places uniformly distributed. For this reason measurement interpolation is applied, exploiting measurements in the neighboring area of the sparsely populated tile.

But in performing the interpolation it is possible to introduce bias from the neighboring measurements. This is mainly due to change of the radio signal path between device and base station, making it subject to influences or interference from other objects. Some areas the signal path does not change significantly, e.g. on country side going along a straight road with open fields on both sides. In such case the signal path will only change due to change in distance to the base station. Conversely, in some areas the signal path changes quickly, e.g. in a city where turning around a corner of a house might yield a completely different signal path. For this reason Paper D investigates what the impact is on mean values when interpolating network performance measurements as a function of interpolation distance, i.e. distance from the tile.

This is done based on measurements of received power and RTT obtained with NetMap in a rural setting and in an urban setting. The interpolation approach applied in Paper D is calculating the mean value of the combined measurements from the tile and the neighboring area. This is a simple interpolation approach, but in principle any interpolation could be applied. It is investigated what the impact is on mean values of RTT and received power measurements when interpolating measurements when increasing interpolation distance.

The results show that there is a clear benefit of performing interpolation in most cases of interpolation distance within the range that is investigated both for received power measurements and RTT measurements. Furthermore, it is concluded that RTT is less sensitive to bias when including neighboring measurements further away than received power measurements. The results are however based on relatively small measurement sets, why the conclusions only apply within the range of 20 to 180 meters in urban setting and in the range of 20 to 300 meters in rural setting. The measurement density was too low to make conclusions on shorter interpolation distances. Furthermore, it was not distinguished at what time of day the measurements were collected.

3.5 Conclusion & Outlook

In this chapter NetMap was presented, which is a crowd sourcing based system for performing and collecting network performance measurements for use in a NPM. In doing this a number of relevant points was considered, such as system architecture, measurement focus and user interaction. Furthermore, we showed from initial measurements that there is great potential in measurements despite them being collected from consumer devices.

We also evaluated an alternative method for estimating achievable throughput with focus reducing resource consumption while not sacrificing measurement value accuracy. This is very relevant as we are exploiting consumer devices that opposite dedicated measurement equipment are limited in available resources. For this reason TOPP was compared with BTC, and was evaluated appropriate as a resource cheap alternative.

Also the received power measurements performed with consumer smartphones were compared with measurements of professional equipment. Here we showed that the low cost alternative is capable of showing detailed effects such shadow fading and path loss.

Finally we addressed the challenge of heterogeneous measurement distribution, when creating a NPM based on measurements collected via crowd sourcing. We evaluated the impact on mean values used in map tiles when interpolating measurements from at different interpolation distances. We showed that there is a clear benefit from interpolating measurements, i.e. there is not introduced bias from the interpolated measurements.

These result combined show that the crowd sourcing approach to network performance measurements is feasible, and has several advantages. Furthermore, NetMap has along the way received a lot of publicity in terms of articles on news sites and in news papers [3] [4] [5] [6] [13], and an invitation as speaker to a conference on network quality [17]. The publicity lead to increased user interest, and thereby installs. As of March 2017 NetMap has had more than 2300 unique installs since April 2016, and has collected more than 2.9 million unique measurements. This lead to interest from several municipalities concerning detailed measurement reports [13], and contact has been initiated with a Danish MNO about collaboration and information sharing. The publicity and general interest from users supported the need for NetMap, and the results that are created from the collected data.

Several future work points related to network performance measurement and crowd sourcing could be mentioned, but here we will just highlight a few. One point is to further investigate how the achievable throughput measurement methods behave on 3G and 4G connections, but also on the upcoming 5G, which promises 1 Gbps rates and 1 ms latency [1]. This will most likely have unforeseen impacts on current measurement methods.

Another point is to further develop the collaboration with MNOs, e.g. in terms of exchanging information. The NPM created from measurements performed with NetMap could be enhanced with information from the internal MNO network. For instance it could be used to check validity of results, or to help describe strange behavior of results. Furthermore, the MNO could also benefit from getting access to detailed measurement results performed on their network. In this way NetMap could act as a diagnostics tool for MNOs that they could ask customers to install to diagnose poor connections.

Another point is more focused on the creation of a NPM. By looking at where and when more measurements are needed, NetMap could use this information to intelligently schedule measurements, depending on where users go and when.

The final highlighted point of future work is less related to crowd sourcing, but related to providing live measurements to services and applications.

With live measurements is meant when a service is in need of performance information about the current connection from the device it is running on. The core of NetMap, i.e. the measurement methods, can without too much effort be integrated in services. This can be done by making a module per measurement method implemented in NetMap that measures the respective parameter to the same measurement point as in NetMap. In that way the experience and infrastructure of NetMap can be reused and potentially make the use of network performance information simpler to integrate in systems.

3.6 Included Papers

This chapter is based on the following papers.

Paper A: NetMap - Creating a Map of Application Layer QoS Metrics of Mobile Networks Using Crowd Sourcing

Paper A is a conference paper which presents NetMap, a crowd sourcing system for performing and collecting network performance measurements. The motivation for collecting the information it to create a network performance map based on actual measurements. Existing network performance maps are usually calculated based on models for signal propagation from base stations. From the theoretical map of signal strength parameters such as throughput and delay are then derived. The measurements are influenced by factors such as varying cross traffic due to user load and movement, device characteristics and load, and variations in the environment. The existing maps will not be able to show these characteristics. The paper presents the architecture of NetMap and describes the system automates crowd sourcing measurements of throughput, round trip time, signal strength and connectivity. From preliminary results collected via NetMap it is demonstrated how indications of varying network load, of the area type the measurement is performed in, and of differences in device performance.

Paper B: Performance Evaluation of Methods for Estimating Achievable Throughput on Cellular Connections

Paper B is a conference paper that evaluates methods for estimating achievable throughput on cellular connections. The background is that most throughput, or bandwidth, estimation methods originally are developed for wired connections, which behave different than wireless connections and cellular connections in particular. This is due to different access schemes and resource allocation, along with the interference from other simultaneous users. The methods estimate achievable throughput as experienced by the application, and not maximum throughput. Achievable throughput is of interest as the goal is to create a network performance map which will show what performance the application can expect. The paper evaluates two different methods; BTC and TOPP. BTC is a simple brute force method for estimating achievable throughput by transmitting as much data as possible within a time duration. For this reason the result of BTC is what TOPP is compared to. TOPP is a probing approach, using far less packets and thereby data. A train of packets is transmitted and the inter packet gap time is logged and used to calculate the estimated throughput rate. The paper evaluates these two methods based on measurements performed on live 3G and 4G cellular networks to compare the methods in a real setting. It is concluded that TOPP achieves satisfactory results when compared to the reliable BTC, but that TOPP will need further tuning of parameters to find the optimal operation settings.

Paper C: Verification of 3G and 4G Received Power Measurements in a Crowdsourcing Android App

Paper C is a conference paper that verifies the accuracy of received power measurements performed from consumer devices by comparing with measurements from professional equipment. This work is motivated by the impact that received power has on resource negotiation with the network, and thereby on other performance metrics. In the paper the evaluation is done based on measurements performed on three Danish live mobile networks, and in four different scenarios. The scenarios cover both urban and highway measurements and measurements performed while walking and driving. The measurements from the consumer devices are performed via NetMap, while professional measurement smartphones and a radio network scanner is used for obtaining the professional measurements. The measurements are compared by calculating mean squared error and cross-correlation coefficient. In comparing the measurements there is accounted for the differences in antenna gain by introducing a power offset. It is concluded that received power measurements obtained from consumer smartphones, e.g. through NetMap, are capable of achieving sufficient accuracy to track shadow fading and path loss. It is shown that slightly higher accuracy is achieved with LTE measurements over 3G measurements, partly due to lower measurement resolution for 3G in the Android API.

Paper D: On the Benefits and Challenges of Crowd-Sourced Network Performance Measurements for IoT Scenarios

Paper D is a journal paper that addresses some of the challenges that arise when working with network performance measurements collected using crowd sourcing for creating a network performance map. The paper specifically addresses the challenges of data being collected from consumer devices, meaning that measurements are only collected where users go. The measurement devices are not dedicated measurement equipment but consumer devices with other applications and user load on networks vary in time and location, meaning that measurement values will vary. For this reason it is not feasible to use individual measurement points in the map. Furthermore, users tend to go some places more than others, meaning that there will be some areas with many measurements and areas will be sparsely populated with measurements. This means that data interpolation is needed to avoid empty areas in the map. In the paper a simple interpolation approach is used where the mean value is calculated of the interpolated measurement values. It is investigated what the impact is on the interpolated values when increasing the interpolation distance from the area with sparse measurements. In the paper the impact is evaluated for RTT and signal strength measurements. It is found that there is a clear benefit in including neighboring measurements when interpolating measurement values in sparsely populated areas. Furthermore, RTT seems less sensitive to bias than signal strength measurements, as introduced by the interpolation.

References

[1] J. G. Andrews, S. Buzzi, W. Choi, S. V. Hanly, A. Lozano, A. C. K. Soong, and J. C. Zhang, "What will 5g be?" *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 6, pp. 1065–1082, June 2014.

[2] K. Højgaard-Hansen, T. K. Madsen, and H. P. Schwefel, "Reducing communication overhead by scheduling tcp transfers on mobile devices using wireless network performance maps," in *European Wireless 2012; 18th European Wireless Conference 2012*, April 2012, pp. 1–8.

[3] Ing.dk, "Aalborg-app afslører dækningen på det danske mobilnet," June 2012. [Online]. Available: https://ing.dk/artikel/aalborg-app-afslorer-daekningen-pa-det-danske-mobilnet-129895

[4] ——, "Studerendes app til overvågning af mobilnet indstillet til produktprisen," October 2012. [Online]. Available: https://ing.dk/artikel/studerendes-app-til-overvagning-af-mobilnet-indstillet-til-produktprisen-132971

[5] ——, "Ny app skal afsløre de sorte huller i mobildækningen," March 2013. [Online]. Available: https://ing.dk/artikel/ny-app-skal-afsloere-de-sorte-huller-i-mobildaekningen-157123

[6] ——, "Derfor kommer der ingen data igennem, selv om mobilen viser fuld dækning," May 2016. [Online]. Available: https://ing.dk/artikel/derfor-kommer-der-ingen-data-igennem-selv-om-mobilen-viser-fuld-daekning/-184484

[7] D. Koutsonikolas and Y. C. Hu, "On the feasibility of bandwidth estimation in wireless access networks," *Wireless Networks*, vol. 17, no. 6, pp. 1561–1580, 2011. [Online]. Available: http://dx.doi.org/10.1007/s11276-011-0364-5

[8] B. Melander, M. Bjorkman, and P. Gunningberg, "A new end-to-end probing and analysis method for estimating bandwidth bottlenecks," in *Global Telecommunications Conference, 2000. GLOBECOM '00. IEEE*, vol. 1, 2000, pp. 415–420 vol.1.

[9] L. Meng, S. Liu, and A. D. Striegel, “Characterizing the utility of smartphone background traffic,” in *2014 23rd International Conference on Computer Communication and Networks (ICCCN)*, Aug 2014, pp. 1–5.

[10] Netflix, “Netflix - internet connection speed recommendations,” March 2017. [Online]. Available: https://help.netflix.com/en/node/306

[11] NetMap, “Netmap application,” March 2017. [Online]. Available: https://play.google.com/store/apps/details?id=nds8.netmap

[12] ——, “Netmap presentation point,” March 2017. [Online]. Available: http://netmap.aau.dk/

[13] Nordjyske.dk, “De sorte huller i mobilnettet skal findes,” March 2016. [Online]. Available: https://nordjyske.dk/nyheder/de-sorte-huller-i-mobilnettet-skal-findes/d6bee85f-2fb0-4c90-8081-94cece9aa9e8/4/1513

[14] M. Portoles-Comeras, A. Cabellos-Aparicio, J. Mangues-Bafalluy, A. Banchs, and J. Domingo-Pascual, “Impact of transient csma/ca access delays on active bandwidth measurements,” in *Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement*, ser. IMC ’09. New York, NY, USA: ACM, 2009, pp. 397–409. [Online]. Available: http://doi.acm.org.zorac.aub.aau.dk/10.1145/1644893.1644941

[15] J. Shi, Z. Guan, C. Qiao, T. Melodia, D. Koutsonikolas, and G. Challen, “Crowdsourcing access network spectrum allocation using smartphones,” in *Proceedings of the 13th ACM Workshop on Hot Topics in Networks*, ser. HotNets-XIII. New York, NY, USA: ACM, 2014, pp. 17:1–17:7. [Online]. Available: http://doi.acm.org/10.1145/2670518.2673866

[16] Stofa, “Stofa fiber bredbånd,” March 2017. [Online]. Available: https://stofa.dk/fiberbredbaand

[17] Teleteknisk Selskab, “Internet quality – more than bandwidth,” June 2012. [Online]. Available: https://teletekniskselskab.wordpress.com/arrangementer/2012-2/internetkvalitet-18-juni-2012/

Chapter 4

Estimating Bus Occupancy Based on WiFi Probes

This chapter will introduce the contributions related to estimating people count in the context of bus occupancy estimation. Specifically we are estimating bus occupancy, i.e. the number of people on a bus, based on WiFi probes. We first present a distributed system for collecting WiFi probes that we have created for the purpose. Next we present our approach to first estimate the number of devices on the bus based on the collected WiFi probes, and from that using probabilistic models to estimate the number of people on the bus. Finally we present a quality indicator that we designed to be attached with the obtained estimated number of people.

This chapter is based on Paper E, and the work is done as part of the Public Transport Optimization use cases taking place in Wolfsburg and Barcelona developed in the BIG IoT project [1]. In these use cases the goal is to optimize public transport both from perspective of the traveler and the public transport provider. The traveler can when given the information about bus occupancy choose a less crowded bus or bus line. The public transport provider can optimize the use of resources, e.g. in terms of assigning extra buses to crowded bus lines as needed.

4.1 Problem Description and Delimitations

People counting is the field of obtaining information about the number of people, which can be used for estimating the load of physical services. Obtaining information about number of people can be done using several different approaches, as presented in Section 2.3, ranging from directly observing to more indirect estimation. A commonly applied method of counting people is based on camera feed supported by computer vision. A less direct approach is using sensors at check points, e.g. counting the number of people passing by. An indirect approach is to count people based on observation of another information type. The different approaches vary in cost and resources needed to realize them in terms of required coverage area and automation.

The information about number of people in an area is highly relevant for many IoT systems as they create the link between digital and physical services. While digital services, such as web services, are loaded by requests from clients, physical services are loaded by the number of people in the area of the service interacting with it. An example of such an IoT system is a travel assistant that supports bus travelers with information such as bus schedules and live information. Examples of such travel assistants are The Intelligent Travel Assistant [3], a personal travel companion [8], Google Maps [4], Inviita [7], and Rejseplanen [9]. All of these systems use schedules and live information about buses and trains to provide users with options for transportation when going from one point to another. These travel assistants could be extended to include information about live occupancy of buses and trains. If a bus is fully occupied the travel assistant should not suggest the user to go with that bus, but rather propose to take another line or wait for the next bus. This could be achieved using information obtained from a people counting system.

There are several challenges in obtaining information about people count. One challenge is related to the required coverage of the people counting in terms of costs and resources. If wide coverage is needed the solution can be very costly in terms of equipment. Furthermore, if continuous estimation is needed an infrastructure must be in place to collect the data when collected. Another challenge is how to evaluate the quality of the obtained information, especially when the people count is obtained based on observing other metrics. In this case there will be inaccuracy connected with the estimated number of people which will depend on the method for obtaining the estimate.

We have chosen to estimate the number of people on a bus based on another type of information, namely WiFi probes. The choice of using WiFi probes for estimating number of people is based on the penetration of WiFi enabled devices [2]. To collect the WiFi probes we have created a distributed system for sensing and collecting WiFi probes where a sensor is placed on a bus. From the collected WiFi probes is obtained an estimated number of devices, and based on this the number of people is estimated. Finally a quality indicator is designed to be attached to the obtained estimate.

4.2 WiFi Probe Collection System

We have created a distributed system for collection of WiFi probes. This system is presented in Paper E, and the architecture can be seen in Figure 4.1. Currently we have deployed live sensors in Wolfsburg in collaboration

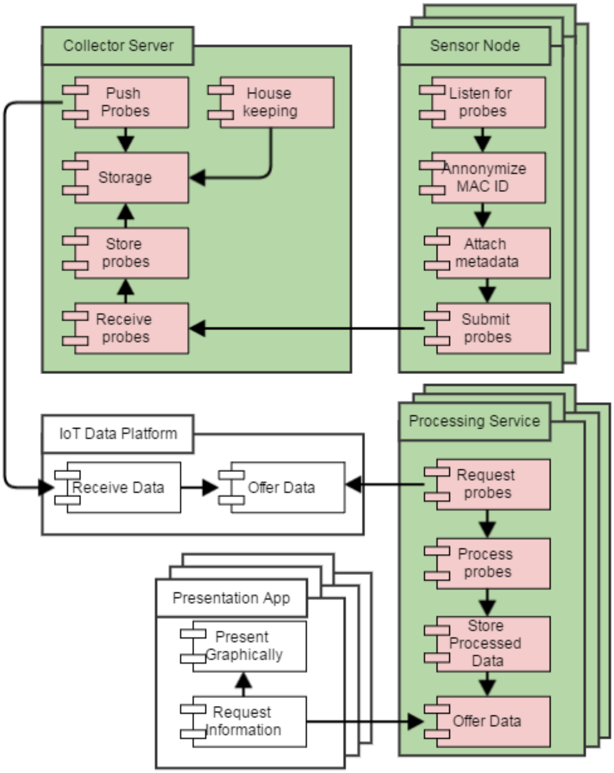


Fig. 4.1: WiFi probe collection system architecture and module interactions (Figure from Paper E).

with Wolfsburg AG in the BIG IoT project [1] and are working on deploying sensors in Barcelona in the same project. We have observed a sensor collecting between a few thousands to more than 60,000 WiFi probes per day, depending on for how long and where the bus is driving with the sensor.

The WiFi probe collection system was designed with the aim of being part of a more general IoT infrastructure, meaning that it should act as a data provider to a bigger IoT platform. In this way the collected WiFi probes can be used in other contexts, e.g. related to smart parking or smart bike sharing. This was inspired by the BIG IoT project [10], where sensors, platforms, services, etc. are operating at different locations and are provided by different entities.

The Sensor Node collects WiFi probes where the sensor is deployed, and anonymizes the MAC address of the collected probes before attaching metadata and submitting them to the Collector Server. The Collector Server stores the probes and pushes them to an IoT platform, where services and applications can subscribe to and access the data. The Collector Server also performs house keeping in cleaning out old collected probes, as over time they lose their relevance.

The anonymization is performed as a one-way operation, meaning that it cannot be reversed to obtain the MAC address. By performing the anonymization at the sensor the user privacy is protected, as no data is stored here. If there is a data breach at the Collector Server the data is anonymized, and will only cover a limited time period.

The Processing Service is where the people count estimation is performed. As it is indicated on the figure there could be several services applying different processing with different goals in mind. This design makes the system highly flexible in terms of scalability and in terms of allowing for different applications of the data.

This WiFi probe collection system is similar to the systems presented in [6] and [5]. Where our approach differs from these two approaches is in obtaining the estimated number of people, which is described in the following.

4.3 Obtaining People Count

With the system in place for collecting WiFi probes, we now estimate the number of people on the bus. The WiFi probes we observe indicates the number of active WiFi enabled devices in the vicinity of the sensor, and from this we want to estimate the number of people on the bus. The challenge in doing this is that we are estimating one parameter based on the observation of another parameter, i.e. the WiFi probes can also originate from devices outside the bus.. So to get from the collected WiFi probes to an estimated number of people on the bus we need to apply some processing.

We do this by first estimating the number of devices on the bus, and then estimate the number of people based on the estimated number of devices. This is presented in the following two sections based on the work done in Paper E, where the full approach is presented.

4.3.1 Estimating Number of Devices

First we create a simple baseline algorithm for dividing the collected probes in two groups; from devices on the bus, and from devices outside the bus.

We assume that probes with low RSSI are further away from the sensor than probes with high RSSI, i.e. possibly outside the bus. Furthermore, when devices are traveling on the bus it is assumed that that the sensor will collect probes from those devices for a duration. Conversely, the sensor will not collect probes for long from devices outside the bus that are quickly passing by. Based on these assumptions we create two threshold values; one for device presence duration, and one for RSSI value.

The output of the baseline algorithm is our baseline estimator of number of devices on the bus. This estimate might contain several wrongly determined devices, i.e. false positives and false negatives, due to the threshold values being too high or too low to detect or filter out certain devices.

For this reason we create an improved estimator by describing a probabilistic model of the baseline estimator and modeling the different components. The components of the baseline estimator are false positives, false negatives, and true number of devices. Based on this model a maximum likelihood estimator (MLE) is derived. The MLE describes the probability distribution of number of devices on the bus based on the baseline estimator. To calculate the MLE we determine the probabilities for obtaining false negatives and false positives through experimental tests.

4.3.2 Estimating Number of People

We now obtain an estimated number of people on the bus based on the MLE, described above, which is an estimate of the number of devices on the bus. We do this by assuming the probabilities of a person having 0, 1, 2, 3 or 4 devices. This distribution for number of devices given the number of persons is compared to the distribution of the MLE. Based on this comparison the most likely number of persons is extracted, which is the final estimated number of people on the bus.

The estimated number of people is compared to the ground truth by calculating the Mean Squared Error (MSE). We obtain the ground truth number of people by visually observing the number of people on the bus while collecting WiFi probes with the sensor. We evaluate the estimated number of people at 10 locations and using 8 different threshold value pairs in the baseline algorithm.

The results indicate that the estimated number of people based on the MLE in general is better than the baseline estimator. In particular a threshold value pair containing medium values for RSSI and device presence time achieves good results.

It should be noted that the results are only achieved based on probes collected on a bus route in Aalborg, and that other bus routes in other cities might require model parameters to be redetermined. Furthermore, the magnitude of the MSE depends on how well the distribution of number of devices per person fit with reality.

4.4 Indicating Quality of Estimator

By using information that is obtained via estimation, such as the estimated number of people on the bus, there is a danger that a bad quality estimate propagates through the system. This may lead to bad recommendations given to the user, or to unreliable decisions made by the system. To avoid this we design and attach a quality indicator to the estimated number of people, which can be used by systems to evaluate how to handle the information in case of low quality.

Taking a parking service as example that provides users with information about which parking lot to park at. Normally when the quality of parking availability information is high, the service automatically chooses a parking lot near the users destination. But if the quality indicated is low, the service can decide not to automatically act on it, but convey this information to the user. The user can then decide to gamble and proceed to a parking lot where the quality of availability information is low. The quality indicator can also be used when fusing information from several sources, e.g. in giving more weight to the source with higher quality.

The quality indicator is created in Paper E, where it is based on the estimation of number of people and not on number of devices. The quality indicator is designed such that it can be used in different settings and domains, just as it was the case for the WiFi probe collection system. The estimator was developed to count the number of people on a bus, but as the maximum number of passengers a bus can carry may vary, the quality indicator is designed to be flexible to this.

The quality indicator is based on the width of the 70% credible interval of the probability distribution of the estimated number of people. The 70% credible interval is between the 15% percentile and the 85% percentile. From this a range is obtained bounded by upper and lower amount of estimated number of people. The size of this range is normalized accordingly to the maximum number of passengers that the bus can take. In that way the quality indicator becomes relative to the maximum range that the estimator can estimate.

When the bus capacity is high and the range of estimated number of people is relatively small, the quality indicator will be high no matter if there are few or many people on the bus.

With the developed quality indicator, attached to the people count information, an application would be able to evaluate the estimate, when also including the bus capacity. For instance if the bus occupancy it estimated to be high but not 100%, then the quality indicator would allow the application to decide if it should recommend the user to take this bus or wait for the next bus.

The advantage of designing the quality indicator is in this way is that it can be easily modified to other buses or trains. It is however limited by the fact that the maximum number people it estimates must be know. This means that if the number of people is to be estimated in a more open scenario, e.g. on the street, a different quality indicator must be designed.

4.5 Conclusion & Outlook

In this chapter we presented a system for collecting WiFi probes that we have created. Based on the collected WiFi probes we created an estimator to obtain the number of people on the bus. This was done by first defining a baseline estimator of number of devices on the bus, which then was modeled probabilistically to obtain an improved maximum likelihood estimator. From the maximum likelihood estimator the estimated number of people was obtained by modeling the number of devices a person carries. The estimator was evaluated based on probes collected on a bus trip, and the estimated number of people was compared to the observed ground truth number of people.

The system design allows for great flexibility of how and where the system is used for collecting WiFi probes. This is especially relevant in typical IoT ecosystems that are highly modular. The system design also allows for usage of the collected WiFi probes, and of the generated information, in different domains than what was presented here.

It can be concluded that it is feasible to estimate the number of people on the bus based on collected WiFi probes. Currently the created estimator is designed and tested based on WiFi probes collected on a bus route in Aalborg. For this reason it might require adjustments if deployed on other bus routes driving in other cities or going through other area types. This is however supported by the approach as it is determined by a set of parameters, that can be adjusted according to the situation.

The designed quality indicator also supports the high flexibility in the overall approach, in that it is scaled to the situation that the estimator estimates people in.

In this contribution we focused on estimating the number of people on the bus by discriminating between probes from devices on the bus and outside the bus. This means that a by-product of the processing is information about the number of devices, or people, outside the bus. Based on this the number of people in the areas that the bus pass through could be mapped. This information could be used by other IoT services, e.g. in determining service load in terms of number of users of the physical service. Other use cases of the generated people count information could also be explored, such as providing predictions for parking lot availability in covered areas, or in social applications providing users with information about which bars to visit to meet the crowd, or maybe to avoid them.

As mentioned the presented system for collecting WiFi probes is highly flexible and can be deployed in many different domains. The sensor does not care if it is placed at a fixed location or is mobile. In any case it will collect WiFi probes from nearby devices and submit the probes to the data collection server. This means that the system is applicable in scenarios such as in airports for monitoring of people flow, or in event venues for crowd safety. Alternatively sensors could be places at fixed locations around the city to support similar use cases.

Currently the estimator for number of people is based on an assumption about the probability of a person having a number of devices. This assumption could be strengthened or modified by performing a survey among a group of bus travelers, asking them about the number of devices they carry and the WiFi state of the devices.

4.6 Included Papers

This chapter is based on the following paper.

Paper E: Accurate and Quality-Aware Bus Occupancy Estimation Utilizing Probabilistic Models for WLAN Probing

Paper E is a journal paper that presents an estimator of bus occupancy and a quality indicator of the obtained estimate. The estimate is obtained based on collected WiFi probes. It was chosen to base the estimator on WiFi probes as this is a low cost and highly flexible solution compared to other approaches for obtaining number of people. The paper presents a system created to collect the WiFi probes, which contains sensors and a data collection server. Based on the collected WiFi probes a baseline estimator is obtained, that estimated the number of people based on a threshold value for device presence duration and a threshold for RSSI values. From the baseline estimator an improved estimator is created by probabilistically modeling the baseline estimator. The parameters needed for the estimator is obtained via experimental measurements where WiFi probes are collected. The improved estimator is obtained using maximum likelihood yielding the estimated number of devices on the bus. To obtain the number of people based on number of devices, the probability of a person carrying a number of devices is assumed. The estimated number of people is evaluated using WiFi probes collected on a bus. The resulting estimated number of people is compared to the observed ground truth number of people on the bus. Finally a quality indicator is designed based on the estimated number of people and the maximum number of passengers on the bus. This is to make it able to adapt to the situation where the number of people is estimated. Results obtained from the approach described in the paper are presented, showing that the approach is capable of estimating the number of people and doing so at a low cost.

References

[1] BIG IoT, "Big iot," March 2017. [Online]. Available: http://big-iot.eu/

[2] Cisco, "White paper: Cisco visual networking index: Global mobile data traffic forecast update, 2015-2020," February 2016. [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.html

[3] J. F. Dillenburg, O. Wolfson, and P. C. Nelson, "The intelligent travel assistant," in *Proceedings. The IEEE 5th International Conference on Intelligent Transportation Systems*, 2002, pp. 691–696.

[4] Google, "Goole maps," March 2017. [Online]. Available: https://www.google.com/maps

[5] M. Handte, M. U. Iqbal, S. Wagner, W. Apolinarski, P. J. Marron, E. M. M. Navarro, S. Martinez, S. I. Barthelemy, and m. G. Fernandez, "Crowd density estimation for public transport vehicles," in *Workshop Proceedings of the EDBT/ICDT 2014 Joint Conference*, ser. EDBT/ICDT 2014, 2014, pp. 315–322.

[6] M. Handte, M. U. Iqbal, S. Wagner, W. Apolinarski, P. J. Marron, E. M. M. Navarro, S. Martinez, S. I. Barthelemy, and M. G. Fernandez, "Estimating crowd densities and pedestrian flows using wi-fi and bluetooth." in *Proceedings of the 11th International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services.*, ser. MOBIQUITOUS 2014, 2014, pp. 171–177.

[7] Inviita, "Inviita," March 2017. [Online]. Available: https://inviita.com/

[8] K. Rehrl, S. Bruntsch, and H. J. Mentz, "Assisting multimodal travelers: Design and prototypical implementation of a personal travel companion," *IEEE Transactions on Intelligent Transportation Systems*, vol. 8, no. 1, pp. 31–42, March 2007.

[9] Rejseplanen A/S, "Rejseplanen," March 2017. [Online]. Available: http://www.rejseplanen.dk/

[10] S. Schmid, A. Bröring, D. Kramer, S. Käbisch, A. Zappa, M. Lorenz, Y. Wang, A. Rausch, and L. Gioppo, "An architecture for interoperable iot ecosystems," in *2016 2nd International Workshop on Interoperability & Open Source Solutions for the Internet of Things*, 2016.

Chapter 5

Enhancing IoT Systems by Including Contextual Information

This chapter will cover the contributions related to enhancing an IoT system using previously presented information types. First we present a smart parking system, which is an IoT system we have created that supports users in finding and paying for parking. In doing this we exploit the functionalities offered by the MOBiNET platform which we also will describe. Next we explore and give recommendations of how to include the information types presented in Chapters 3 and 4.

We created the intelligent parking assistant system as a use case demonstrating the use of the MOBiNET platform [8] functionalities. The system was created in collaboration with North Denmark Region, providing access to parking lot information, and GateHouse, providing the back end system and user and billing management. This chapter is based on Paper F.

5.1 Problem Description and Delimitations

Systems have in recent years moved from providing simple functionalities or services to the users, to being smart systems making intelligent decisions about what to provide to the user, based on relevant information. But a system is only as good as its user base, in terms of revenue, why it is important that a system covers more areas, e.g. geographically or functionality wise. This can be done by developers seeking out relevant data sources to expand the system, and thereby the user base. However, this might be a costly and time consuming affair as the system potentially needs to be adjusted according to each service or information source. In this case the system operates as an isolated system, only interacting with specifically selected entities.

This is a well identified problem, and the solution to get out of isolation is to utilize service discovery. With service discovery the system can be prepared to automatically discover and use new data sources. This is supported by a wide range of platforms, implementing various approaches for performing service discovery, such as presented in Section 2.4. Different choices of service discovery have different pros and cons, making the choice of approach a trade-off. Typically the trade-off is between difficulty of use and functional complexity. The service discovery approaches that are easy to use are typically offering less complex capabilities, i.e. service discovery based on simple filtering rules. More complex service discovery approaches typically employ semantics to enable fuzzy searches for service capabilities and data sources, i.e. service discovery based on semantics and ontologies.

The MOBiNET platform employs a service discovery approach based on a range of different filtering options [10]. This is not ground breaking in it self, but in MOBiNET the full range of platform functionalities are implemented and made available. The MOBiNET platform focused on mobility services in the domain of intelligent transportation services (ITS). The MOBiNET platform architecture is presented in Figure 5.1, and the functionalities are described in detail in Section 5.2.

In the MOBiNET project we are tasked to realize a use case that demonstrates the MOBiNET platform functionalities, by creating a smart system. Conceptually the system should demonstrate the advantages of utilizing the MOBiNET platform, particularly related to the ease of expanding the system to cover more cities or countries. This should be done by utilizing the MOBiNET core services; Service Directory, Billing Manager, and Identity Manager. The goal of MOBiNET is to create an ecosystem where information and services are discovered and used automatically by different applications and services. In this connection we have chosen to work with the ITS sub-area of smart parking. This is relevant due to the high fraction of moving vehicles in the city that are searching for parking spots [2], and the congestion that this entails [14], [3]. For this reason it is critical that users easily can find and pay for parking, and thereby reduce the time spent on this.

We have created a smart parking system that provides the user with live parking availability information, and automatically initiates and pays for parking sessions based on the user location. The system utilizes the MOBiNET platform service discovery to automatically discovering parking information sources based on the location. This system is made highly expandable in terms of covering more cities and parking lots thanks to MOBiNET functionalities.

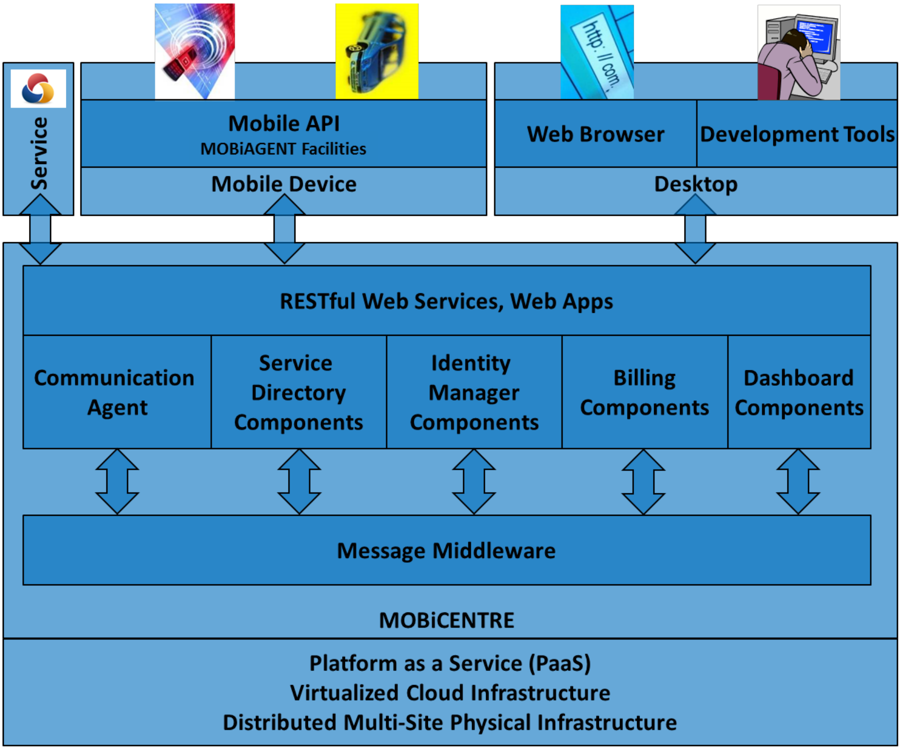


Fig. 5.1: The architecture of the MOBiNET Platform [15].

5.2 Intelligent Parking Assistant System

As a use case demonstrating the MOBiNET platform we have developed the Intelligent Parking Assistant system. This is a system capable of supporting users while parking, in terms of providing live information about parking lots and automatically initiating payment for parking sessions. The system has been tested in a live test in Aalborg by approximately 40 users, and is currently up for user testing in Trikala, Greece. For a demonstration of the system please refer to [17] and [9]. In the following we will describe the system which is presented in Paper F.

The goal of the Intelligent Parking Assistant system is to make finding and paying for parking more efficient, thereby saving time and reducing pollution. The system consists of several components; Intelligent Parking Assistant application, Parking Information Service, Parking Manager, Map Matching Server, and MOBiNET platform, as shown in Figure 5.2.

The core of the system is the client application, the Intelligent Parking

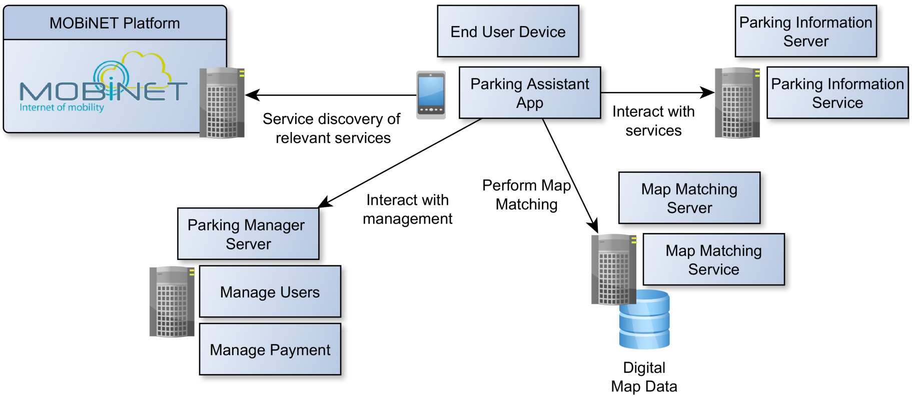


Fig. 5.2: Main components of the Intelligent Parking Assistant system and the main interactions.

Assistant [6], that users install on their smartphone. In the application the user logs in, to associate with an account registered on the Parking Manager Server. In the application the user can enter several cars in terms of license plate. From the application the user can choose to be guided to an available parking lot nearest to his destination. Alternatively he can choose to see a map of all parking lots in the area, including indication of availability. By selecting a parking lot the user is presented with information about services and prices but also live availability information of the chosen parking lot. From the car park information page the user can get turn-by-turn route guidance to the parking lot.

The Parking Information Service is providing information about parking lots, in terms of location and geographical boundaries, live availability, price, services, and other. The service can cover all parking lots in a city or country or just a subset of parking lots, depending on setup and on availability of parking lot information. As an example of adding another Parking Information Service is done in Trikala, Greece, where a local service provides information about local parking lots. This service will be discovered by using the client application on a device in and around Trikala.

The Parking Manager is managing the user accounts and keeps track of parking sessions. It is accessible via an API for the application, but it also provides a web interface for users to sign in and manage their accounts. It is here that the user registers his payment card such that automatic payment can be performed by the application. The Parking Manager keeps track of parking sessions enabling it to provide additional functionalities to the system. In case of the application crashing or the device running out of battery the parking manager will serve as a back-up method for users to terminate active parking sessions.

The Map Matching Server supports the application in automatically initiating parking sessions based on location information from the user device. This is done by the client application logging the location of the device, and thereby the car, for a duration until the car is parked. When this happens the array of locations are submitted to the Map Matching server, that performs map matching [1] of the location trace of the device, location accuracy, a digital map of the road layout and the geographical boundaries of the parking lot. The output is a probability of the device, or car, being stopped at an exact location, i.e. distinguishing between being inside a parking lot, or being parked on the road next to it. The result is returned to the application that evaluates the probability and decides whether to initiate a parking session or not.

In case that the Map Matching server does not have a digital map of the road layout available the application will fall back to a simpler solution, only relying on current location of the device and on the geographical boundaries of the parking lot. Here also the accuracy of the device location is taken into account in evaluating if the device, or car, is stopped inside the parking lot.

Several functionalities of the MOBiNET platform are used in realizing this system. Most important is the service discovery of the MOBiNET Service Directory. This allows the application to discover relevant parking information services based on the location of the user device, service type and based on the needed parking information data format. Other supported discovery criteria are QoS level, pricing, and service domain category. To make a service discoverable through MOBiNET, a service developer or owner has to publish a service description in the service directory. This service description contains information about the service endpoint, interfaces, service type, data input and output, coverage area, pricing, and many other parameters. In the Intelligent Parking Assistant the service discovery is used with location information, service type, and data format.

Additionally MOBiNET provided functionalities used by the system are Identity Manager and Billing Manager. The Identity Manager is used to authenticate the identity of parking providers when the users performs roaming parking. In this scenario a parking session is initiated at the parking manager that the user is visiting, based on this parking manager being authenticated through the Identity Manager. The Billing Manager is used to perform business to business billing such that the user pays to his account managing parking manager, which then pays the ohter parking manager that the user visited.

Also the Data Format Catalog, which is a sub-component to the service directory, and the Dashboard are used, however mostly in the development phase. The Data Format Catalog supports developers with descriptions of the data formats used and provided by services described in the service directory. The Dashboard provides the general access portal to the MOBiNET platform.

With the implementation of the Intelligent Parking Assistant system we validate the MOBiNET platform functionalities, by showing the advantages of using the MOBiNET platform. Specifically the service discovery and the business to business billing aspect makes the MOBiNET platform approach relevant when wanting to expand the system coverage and user reach.

Our smart parking system is different from other smart parking systems in that we do not require parking lot owners or parking information providers to make a dedicated registration of parking lots in our system. Instead we propose to publish the information in the MOBiNET Service Directory, where our system automatically can discover the information, just as any other smart parking system using MOBiNET. The only requirement is that the service offers the same parking data format as what is currently used by our smart parking system. However, in the case of different parking format offered a translation service could be employed.

Conversely, other smart parking systems make ad-hoc agreements with parking lots about obtaining live information and/or supporting payment [7] [12] [11]. This can be very time consuming and costly in terms of development work.

We will now present proposals for how the MOBiNET service discovery can be enhanced in the future by using the information types network performance and people count. Furthermore, we will also propose how to make the smart parking system stand out from similar solutions besides the use of the MOBiNET platform, also based on including the information types network performance and people count.

5.3 Envisioned Information Exploitation

To link with the overall problem statement we will now present how we envision service discovery and the smart parking system could be enhanced in the future by including the previously described information types network performance and people count.

5.3.1 Exploiting People Count Information

In the service discovery and selection offered by MOBiNET we envision people count information can be used in the service selection phase. This could be done by selecting services based on their capacity and based on information about people count in geographical areas. This assumes that service developers describe an availability parameter in the service description that is published in MOBiNET Service Directory. Provided this can be done, it could serve as a load balancing function for services that are fully loaded, and therefore should not be discoverable. This would require that the people count information is available to the MOBiNET Service Directory. This functionality can only be supported where the information is available. For this reason the opportunistic approach to collect the information is relevant as large areas can be covered at low cost.

People count information is highly relevant to the Intelligent Parking Assistant because it operates in the domain of parking, which naturally is affected by parking lot availability, traffic and users. Here people count information is referring to number of people in an area, but it can also be used for extracting information about people flow, meaning information about where people are going to and from.

One usage of people count information is to evaluate if there are many or few people in the area of the parking lot. If there are many people in the area, the probability of the parking lot being fully occupied is high. This could possibly be used to estimate availability of a parking lot, even when there are no sensors providing live availability information about that parking lot. Prediction of parking spot availability is a research area with many contributions, however most focus on either camera based approaches [16], or model based prediction trained on observations [4], [13], [5]. Another usage of the people count information is assessing if many people are going towards the same area, which the user then possibly have to compete with to get a parking spot. If there is a general tendency that people are moving towards the area where the user is heading, the probability of the parking spot being taken is high. Conversely, if there is a general tendency that people are moving away from the area, then the probability of the parking spot being taken is low.

These usages could further be enhanced if the people count information contains an indication of quality, such as presented in Chapter 4. Based on this if the quality is above a certain level, e.g. 90%, the Intelligent Parking Assistant could use the information as usual. But if the quality is below that level, it could present the user with a warning along with the information or decision. In that way the user can take precautions in case of uncertain or inaccurate information, and he can choose the best course of action.

5.3.2 Exploiting Network Performance Information

Service discovery and selection in MOBiNET is done based on different parameters, where a key parameter is the location, but network performance information could also be utilized. This could be done by in the service description defining requirements to network performance, e.g. in terms of delay or throughput, or to the stability of these. Based on the input location the network performance in the area could be extracted, which then could be compared to requirements described by the service descriptions. In that way only the services that can be properly interacted with in the area, based on the network performance, will be discoverable. This would also ensure faster servicing of requesters because the occurrence of unexpected poor network performance could be reduced, provided that the network performance information is reflecting the state of the network. For this reason crowd sourced network performance measurements are relevant as they are actual measurements, perform at the geographical locations where the information is needed.

Network performance information is also relevant to the Intelligent Parking Assistant system due to the distributed nature of the system. The performance of the network is influencing the system performance because information is communicated between the client application and various back end servers. The network performance will have an impact on system performance in terms of general system responsiveness, and in terms of having the most up to date information when needed. By including network performance information the Intelligent Parking Assistant could prepare for entering areas with poor network performance, e.g. by pre-loading needed information from relevant services. Furthermore, it could also present the user with a warning of poor network performance to ensure user expectations to system responsiveness are adjusted.

5.4 Conclusion and Outlook

In this chapter we have presented a smart parking system that we created. The goals of creating this system was to present the functionalities of the MOBiNET platform, and in particular how smart systems are easily expanded when utilizing the MOBiNET platform services. We described how we include information relevant to parking and to the current scenario of the user. We also described how MOBiNET service discovery functionalities are exploited by the application to automatically discover and use new information services. We proposed how the MOBiNET service discovery approach can be enhanced by including information about network performance and about people count. This relates to how the information is included in the internal service discovery process in the MOBiNET Service Directory. We also proposed how the smart parking system can be enhanced based on the two information types. This relates to how services are interacted with, and how live parking availability information is enhanced.

The presented smart parking system could be developed further by integrating it in a multi modal travel assistant, supporting a park-and-ride service. A park-and-ride service is for users commuting to cities that park at the edge of the city and use public transport the last stretch. In doing this the full range, or a subset, of functionalities in the Intelligent Parking Assistant could be included. The goal is to provide the user with a one-stop for all traveling needs, i.e. where to park and the availability of parking lots, public transport options and schedules, and payment of all used services.

It was mentioned previously that people count information could be used to estimate parking lot availability. This would however require some experimental tests where people count information and parking lot availability information is correlated. This has some very interesting prospects as parking lot owners could offer availability information without investing in expensive counting sensor solutions.

5.5 Included Papers

This chapter is based on the following paper.
Paper F: Intelligent Parking Assistant - A Showcase of the MOBiNET Platform Functionalities
Paper F is a conference paper that presents the Intelligent Parking Assistant system as a use case of the MOBiNET platform functionalities. The goal of this system is to demonstrate how systems can be made that automatically exploits available services by utilizing the MOBiNET platform. The different components of the smart parking system are presented in terms of client application, back end components, information providing service and MOBiNET platform services. The key functionalities of the client application are presented in detail, including automatic initiation and payment of parking sessions based on geographical location sensing supported by map matching. Regarding the MOBiNET platform the used functionalities are described, mainly focusing on the service discovery functionality of the Service Directory. The back end components that support the client application during operation are also described. One back end component is the parking information service that support the application with information about parking lots in terms of live availability and geographical location information. Another back end component is the parking manager that supports the application in managing parking sessions, and it also manages user accounts. A final back end component is the map matching server to which the client application submits location samples to. Based on the location samples, and based on digital road maps, performs map matching to provide a probability of the device, or car, being at a specific location. Finally the use cases of the Intelligent Parking Assistant system are described, highlighting the advantages of using a smart parking system for the user. The paper presents how MOBiNET enables the creation of smart systems that automatically are expanded based on availability of new services.

References

[1] N. Agerholm, H. Lahrmann, B. Jørgensen, A. K. Simonsen, and S. Tøfting, "Full-automatic parking registration and payment - in principle GNSS-based road pricing," *10th ITS European Congress*, June 2014.

[2] M. Caliskan, A. Barthels, B. Scheuermann, and M. Mauve, "Predicting parking lot occupancy in vehicular ad hoc networks," in *2007 IEEE 65th Vehicular Technology Conference - VTC2007-Spring*, April 2007, pp. 277–281.

[3] A. L. Fauconnier and E. Gantelet, "The time looking for a parking space: strategies, associated nuisances and stakes of parking management in france," *European Transport Conference (ETC)*, 2006.

[4] P. Ioannou and Y. Zhang, "Intelligent driver assist system for urban driving," in *2016 Digital Media Industry Academic Forum (DMIAF)*, July 2016, pp. 128–134.

[5] J. S. Leu and Z. Y. Zhu, "Regression-based parking space availability prediction for the ubike system," *IET Intelligent Transport Systems*, vol. 9, no. 3, pp. 323–332, 2015.

[6] LMMApps, "Parking assistant," March 2017. [Online]. Available: https://play.google.com/store/apps/details?id=dk.mobinet.parkassist

[7] MOBILPARKERING, "MOBILPARKERING," March 2017. [Online]. Available: https://mobilparkering.dk/

[8] MOBiNET, "MOBiNET - Internet of Mobility," November 2012. [Online]. Available: http://mobinet.eu/

[9] ——, "Parking assistant demo," March 2017. [Online]. Available: http://mobinet.eu/?q=content/parking-services-ps-demo-use-case

[10] U. Noyer, T. Schlauch, P. Cercato, and L. Mikkelsen, "Mobinet – architecture overview of an innovative platform for european mobility services," *22nd ITS World Congress, Bordeaux, France*, October 2015.

[11] ParkMan, "ParkMan," March 2017. [Online]. Available: http://www.parkmanworld.com/

[12] ParkMe, "ParkMe," March 2017. [Online]. Available: https://www.parkme.com/

[13] T. Rajabioun and P. A. Ioannou, "On-street and off-street parking availability prediction using multivariate spatiotemporal models," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 5, pp. 2913–2924, Oct 2015.

[14] R. Ranjini and D. Manivannan, "A comparative review on car parking technologies," *International Journal of Engineering & Technology*, vol. 5, April 2013.

[15] T. Schlauch, D. Beckmann, I. Passchier, N. Zahariev, and L. Mikkelsen, "MOBiNET - an innovative approach for a European-wide ITS service platform," *10th ITS European Congress*, June 2014.

[16] I. V. Sukhinskiy, E. A. Nepovinnykh, and G. I. Radchenko, "Developing a parking monitoring system based on the analysis of images from an outdoor surveillance camera," in *2016 39th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, May 2016, pp. 1603–1607.

[17] K. Zountouridis, “Parking assistant demo video,” March 2017. [Online]. Available: https://www.youtube.com/watch?v=UBc5h8q8tqo&feature=youtu.be

Chapter 6

Conclusion and Outlook

6.1 Conclusion

In this work we have explored how to enhance IoT systems by exploiting opportunistically collected information about network performance using crowd sourcing, and information about people count collected passively from WiFi communication. In doing this we attempted to answer the main problem statement and the three listed subquestions from Section 1.4:
How can IoT systems be enhanced by utilizing opportunistically collected information about network performance and information about number of people in a limited geographic area?

To obtain network performance information we created NetMap, a tool for performing and collecting network performance measurements using crowd sourcing. The NetMap application automatically performs measurements periodically and submits them to a data collection server. This means that minimal effort is needed to create a network performance map (NPM) based on actual measurements on real networks. Furthermore, two measurement methods for estimating achievable throughput were compared with the goal of lowering the data used for obtaining this parameter. This goal was achieved using the measurement method TOPP. Also the received power measured using smartphones was verified by comparing with professional measurement equipment. This showed that the received power measurements provide a sufficient level of detail to track shadow fading and path loss. Finally we addressed the issue of having areas sparsely populated with measurements when creating a NPM. This was done by investigating the impact of interpolating measurements in areas with few measurements. This contribution shows that it is feasible to perform and collect network performance measurements using crowd sourcing, and that a NPM can be created from the collected measurements. The achieved results are however only obtained in limited scenarios, so to further support the results an analysis should be performed based on measurements collected on a larger scale.

To obtain the number of people in a limited geographic area we collected WiFi probes and applied probabilistic modeling to obtain an estimated number of people. This was done in the scenario of bus occupancy estimation, i.e. counting the number of people on a bus. To collect the WiFi probes a system was created where sensors paced on buses passively collect WiFi probes, anonymize the unique identifiers and submit the probes to a probe collection server. The system is designed to be modular and highly scalable to enable the addition of many sensors, but also enabling the addition of additional services providing other processing methods. We created a maximum likelihood estimator (MLE) of the number of people by using probabilistic modeling of the components of a baseline estimator. The results showed that the MLE achieved higher accuracy than the baseline estimator, when compared to the ground truth number of people on the bus. Based on the estimated number of people we developed a quality indicator of the estimate. The motivation is to give systems a chance to stop low quality estimates propagate through the system. The resulting quality indicator is a general solution that can be applied by other estimation approaches in the same scenario, which is relevant when performing data fusion. This contribution shows that bus occupancy can be estimated effectively based on WiFi probes. Furthermore, it also demonstrated how the information can be assessed in terms of quality, which is useful for applications. The obtained estimate is however limited by the distribution of the number of devices per person being an assumption. To further support this assumption a survey could be performed among bus travelers. This contribution is done in connection with the BIG IoT project, where this bus occupancy estimation will be part of several use cases related to public transport optimization.

To show how to exploit the information types network performance and people count information in a real IoT system, we created the intelligent parking assistant system. This system supports the user with live availability information and helps him in finding and paying for parking. Currently it does this based on information about the user location and movement. This showed that it is possible to create a smart system that based on context information can adapt automatically, in terms of selecting relevant information sources. The system was created in connection with the MOBiNET project to demonstrate the capabilities of the MOBiNET platform, especially with focus on the service discovery functionality offered by the MOBiNET Service Directory. The intelligent parking assistant system demonstrated how systems can easily be expanded in terms of covering more cities and countries by utilizing the MOBiNET services. We proposed how network performance information and people count information can be used to enhance the current service discovery offered by MOBiNET, by enhancing the internal discovery process. We also proposed how to enhance the intelligent parking assistant system by utilizing the two information types. Specifically, people count information can be used to enhance the live parking availability information or even add it to parking lots without sensors as a low cost alternative. Network performance information can be used both in relation to scheduling of data download, and in relation to preparing the application for longer response times from distributed system components. This contribution shows how to create a smart IoT system that makes automatic service discovery based on context information. Furthermore, it is presented that the system has the potential of being further enhanced by including the information types network performance and number of people. The latter point is however only proposals and must be realized to be verified.

6.2 Outlook

Several of the results presented in this thesis invite for further research. We will now highlight key areas for further research.

Regarding further research in the area of crowd sourcing of network performance measurements the creation of a network performance map (NPM) is a key point. There are several aspects to this, e.g. how to present network performance values in a meaningful manner. This includes how to present the different variations in network performance that could occur in a small geographic area. This could for instance be in terms variations according to the day of the week and according to the time of day. But also how to represent the mean and variance such that it makes sense to users. Another point is how to further collect the needed measurements, specifically how to make intelligent scheduling of measurements according to the need of the NPM. Another area of further research related to the network performance estimation is related to the emergence of new connection technologies such as 5G. This is both in terms of how the new technology affects the measurement methods, but also in terms of ensuring that the measurement methods can track the high performance offered by the technology.

One area of further research related to estimation of number of people is to utilize the information in other domains than public transport optimization. This could for instance be to utilize the information in determining parking lot availability, or to estimate the number of people in different areas of sport arenas. In these cases the estimation approach would have to be modified because currently the number of people on the bus is estimated. Another area of further research is how to utilize information about devices outside the bus. Currently the focus has been on determining which WiFi probes originated from devices on the bus. In this process a by-product is WiFi probes from devices outside the bus. This could have several usages such as identifying people flows in the city where the bus operates, or simply counting the number of people in the areas the bus pass though. A third area of further research is to explore the area of data fusion by taking into account the quality indicator.

Related to the smart parking system the key areas for further research are described in Chapter 5, but additional areas can be mentioned. Currently the intelligent parking system utilizes the MOBiNET platform, to discover and use new parking information services. The system could be modified to exploit other service discovery approaches provided by other platforms. Furthermore, it could be explored how people count information could be exploited as a general context information to cyber physical systems that are influenced by the number of physical users.

A final point of further research to highlight is to explore how to combine the information types network performance and people count. This could be done related to the creation of a NPM, to help explain why a certain level of network performance is experienced at certain times. This could for instance be based on information about number of users in an area, correlated with the measured network performance in the same area at the same time. If such a link can be made it can be explored if network performance fluctuations can be predicted based on people count or people flow information.

Part II

Papers

Paper A

NetMap - Creating a Map of Application Layer QoS Metrics of Mobile Networks Using Crowd Sourcing

Lars M. Mikkelsen, Steffen R. Thomsen, Michael S. Pedersen, and Tatiana K. Madsen

The paper has been published in the
Proceedings of 14th International Conference, NEW2AN, and 7th Conference, ruSMART pp. 544–555, 2014.

Abstract

Based on the continuous increase in network traffic on mobile networks, the large increase in smart devices, and the ever ongoing development of Internet enabled services, we argue for the need of a network performance map. In this paper NetMap is presented, which is a measurement system based on crowd sourcing, that utilizes end user smart devices in automatically measuring and gathering network performance metrics on mobile networks. Metrics measured include throughput, round trip times, connectivity, and signal strength, and are accompanied by a wide range of context information about the device state. The potential of the NetMap approach is demonstrated by three usage examples.

A.1 Introduction

In recent years mobile networks have seen a huge increase in data transferred (and is estimated to continue to increase [4]), while other types of network usages drop (SMS and speech [6]). This trend is highly associated with the large increase in the amount of smart devices, such as tablets and smartphones, which are getting more and more powerful in terms of functionalities, processing power and connection capabilities. Along with the more powerful devices, another reason for the high increase in network traffic is based on the services being used, such as video streaming (estimated to grow to more than 69% of all consumer Internet traffic [4]), music streaming (Spotify, Grooveshark, etc.), constantly updating social networks (Facebook, Twitter, Instagram), and in general network enabled services. At the same time Internet Service Providers (ISPs) are deploying faster network technologies [2], e.g. HSPA+ and LTE/4G, enabling higher data transfer rates and QoS for the users of the networks.

Users are getting accustomed to have high speed networks and capable devices. However, when degradation in performance is experienced, it might not be easy for a user to identify a factor causing performance drop. The device he is using might have poorer capabilities than expected, such as poor antennas, network chip, or processing power. The network might be poorly configured or might lack high performance capabilities in certain areas. Lastly the service might not be scaled to the amount of users using it, or it might be slow due to momentarily high load. Regarding the service, there are usually ways of estimating if the service is experiencing sub par performance at the moment [12], but generally the user has no influence on any of the performance impediments of the device or network. To make sure that he receives the best possible performance, he need to compare the performance of his device on his ISP network with other devices, or with other ISP networks in the area.

This is where a proposed system called NetMap [8] comes into play. NetMap is a mobile network performance measurement system based on crowd sourcing, which continuously gathers information about the performance of the network connection from the device to the backbone of the connected network. Based on the information gathered via NetMap a geographical Network Performance Map (NPM) of mobile networks is generated while also mapping the performance of the devices.

A.1.1 Advantages of NetMap

Today most ISPs have made coverage maps available to their users, where it is possible to see an estimated signal strength in an area and an estimated throughput speed [10]. These maps are based on theoretical calculations, and give a static image of the expected coverage, but true experienced performance may vary due to local conditions such as hills, trees and houses, and might be affected by dynamic factors such as network load.

Here NetMap has a clear advantage, as it offers a NPM based on actual measurements on existing networks, and by using actual end user devices in real end user scenarios. The NPM shows what network speeds to expect, based on real empirical and continuously updated data, and at the same time it provides a more realistic image of what the end user can expect as the measurements are performed with devices similar to his. Another advantage of generating a NPM using NetMap, is the way the data is generated and gathered: by using crowd sourcing. It would be very time consuming and require a large amount of resources, including equipment and man labour for an ISP to create this type of NPM. Furthermore, the client side software, i.e. an app, can be distributed by using existing infrastructure, such as Google Play or Apple App Store. Additionally, a wide range of metadata is extracted on the state of the device while the measurements are being performed, e.g. location, movement speed, battery level, connection type, etc., which can help in understanding the cause of fluctuations in the measured performance.

A.1.2 Existing Solutions

In [7] an approach for making a network coverage map based on crowd sourcing is proposed, which in many ways is similar to NetMap, but only measures signal strength, resulting in a network coverage map being constructed.

Open Signal [9], available as an Android app, is aimed at measuring physical connection metrics by mapping cell towers, signal strengths and WiFi access points, and the results are presented in a coverage map. Furthermore it is possible to manually perform throughput and ping measurements in the app.

Both solutions use crowd sourcing to perform and gather measurements on mobile networks, and their focus is on physical connection measurements, while the main focus of NetMap is to perform application layer network performance measurements, and creating a NPM.

A.1.3 Measurement Approach in NetMap

In Figure A.1 the overall measurement approach used in NetMap can be seen. The concept relies on the ISP networks being connected to an Internet exchange point, and the measurement point being located on a neutral network, also connected to the Internet exchange. By neutral network is meant a non-mobile ISP network, where no mobile devices are directly connected using mobile technologies like 3G, 4G etc..

In Denmark an example of such a neutral network is the Danish Research Network [5], which does not offer any mobile connections, and is connected to the danish Internet exchange. Looking at Figure A.1, by comparing measurements from Clinet #1 and #2, the network and the device performance will be compared, as the measured entity is the connection starting at the device and ending at the internet exchange point.

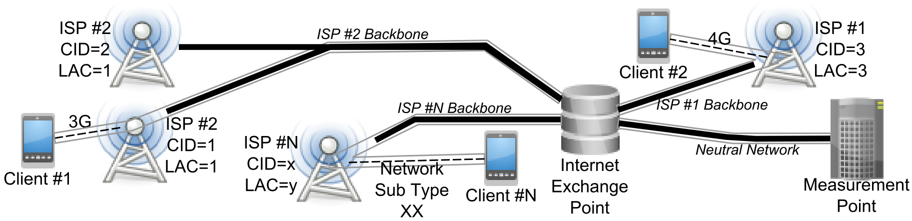


Fig. A.1: Concept of the measurement approach in NetMap (measurement traffic is marked with gray). For description of CID, LAC and Network Sub Type see Section A.3.2.

As the measurements are performed via an application installed on a mobile device like any other application, the measured performance is comparable with regular network usage of end users, as it is not a dedicated device, or a dedicated network interface, while other applications running will also consume the resources available on the device during measurements. Furthermore, due to the diverse market of smart devices, the measurement devices will vary in quality, and thereby vary in quality of connectivity hardware.

Both these issues are addressed by gathering metadata on the device. With the metadata it is possible to understand the situation the measurements are performed in, both in terms of context of the device and the state of the device, as it allows various filtering and sorting of the measurements.

A.1.4 Requirements to NetMap

In order to give NetMap a higher chance of both getting more users and keeping users longer, while also increasing the chance of generating results that will have an impact, some requirements are defined.

REQ1 **Low resource usage** of measurement methods (data) and application in general (power, memory, etc.).

REQ2 **Unbiased measurements** as there are multiple mobile networks it is important that the measurements are not biased towards any network.

REQ3 **Relatable metrics** as the user is not necessarily a technical person the measured metrics need to be understandable for this type of user as well to maintain interest/motivation.

REQ4 **Transparency of measurement methods** is important such that it can be verified that the results are obtained by using correct methodology.

REQ5 **Simple to use system** e.g. by making it fully automatic is important as if the user needs to go through too much hassle to perform or submit measurements he will be inclined to not use it.

REQ6 **Privacy** of the user is a vital point, as a lot of metadata is gathered about the context of the device, and it should not be possible to identify a single user from this.

A.2 General System Functionality/Architecture

In Figure A.2 the main actors, components and entities of NetMap can be seen, and will in the following be described.

NetMap App: The client side software from where the measurements are initiated, scheduled, and results submitted to Collection Point. NetMap App is the front end, as seen from the end user perspective, giving the user basic control such as start/stop of measurements, selecting measurement types to run, setting maximum data usage, and other settings. NetMap App also offers immediate feed back from the measurements, i.e. latest results or aggregated results.

Measurement Point: The end point of all measurements performed in NetMap, which is responsible for performing the measurements requested from NetMap App. This is the only truly critical point in the design and implementation of NetMap, since if it is not properly scaled to accommodate all requests the measured values will be affected. In that case the measurements will not show the true state of the network, but rather the state of the resources of the server. Therefore initiatives, such as multiple servers and

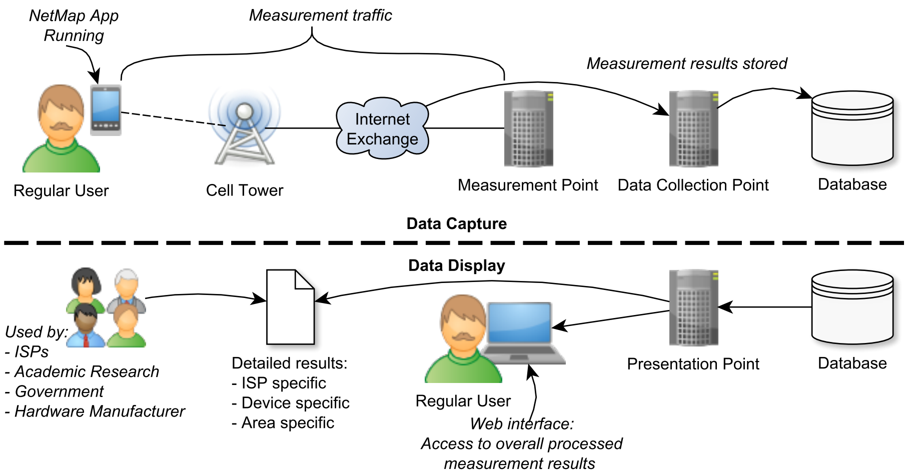


Fig. A.2: Overview of the entities and data flows in NetMap.

load balancing, and maximum allowed simultaneously active measurements, are taken to ensure scalability of the system.

Data Collection Point and Database: Measurement results are submitted to Data Collection Point, and when results are received they are stored in the database. This setup allows for version checking of result objects before adding them to the database, so if a measurement method is updated and additional result fields are added this can be handled before adding the results to the same table as the previous version of the same measurement method.

Presentation Point: The entity responsible for processing the data in the database and presenting it according to the specific purpose. For the NPM purpose, Presentation Point extracts the data from the database and performs some preprocessing to generate a reduced dataset which is the NPM. The reduced dataset is then made available as an API, that can be used by webpages and applications that offer NPMs to end users.

A.2.1 End User Controls

As mentioned with NetMap App, the end users have some options for basic control. This is due to the fact that having an application constantly running and performing measurements could potentially drain the available resources on the mobile subscription. Therefore the user is able to control the data usage by setting an upper limit, selecting which measurement types to run as some use more data than others, and by setting the time between each measurement run.

A.2.2 End User/Crowd Motivation

The outcome of NetMap relies heavily on end users installing the app and performing measurements, why motivation of users to participate is a key aspect of the system concept. Currently users are motivated to participate only based on measurement results of NetMap, which can be used to check if they see expected or sub par performance. In an initial 4 week trial period, where end users only had access to view their own measurements, approximately 1500 users chose to participate, which shows that users are motivated as there is a general interest in evaluating network performance. However, based on the limited format of the output to users, it is likely that participants were mostly technical savvy persons, why additional motivation is needed to reach a broader user base.

A.3 Metrics and Measurement Methods

The measurements are scheduled according to a periodic scheme, where measurements are performed at a fixed interval of 15 minutes, which can be changed by updating the app. By changing the interval between measurements, i.e. changing the frequency of measurements, the maximum load at Measurement Point can be controlled according to number of active users of NetMap. Following are the measured metrics described along with arguments for why they are relevant.

A.3.1 Application Layer QoS Metrics

Bulk Transfer Capacity (BTC): BTC, or throughput, as defined by the Internet Engineering Task Force (IETF) [11], is the average amount of data that can be transmitted over a link per second, $BTC = data\_sent/elapsed\_time$, also known as bit rate or bits per second (bit/s). As the measurements are performed at the application layer, data represents the bits received excluding headers or overhead, and time is measured as the time between the first received bit and the last. The throughput is measured for both uplink and downlink using TCP, and each measurement is performed for 15 seconds. In Figure A.3 the method for measuring throughput is illustrated, indicating that it is a simple procedure where as much data as possible is pushed through the connection making it rather expensive in data usage.

It is interesting to know the throughput a connection can deliver, especially in connection with content based applications such as streaming or cloud services. Furthermore, as the throughput is the most common performance metric used to describe a connection, both by users and ISPs, it is the metric giving the best perception of how well a connection or network

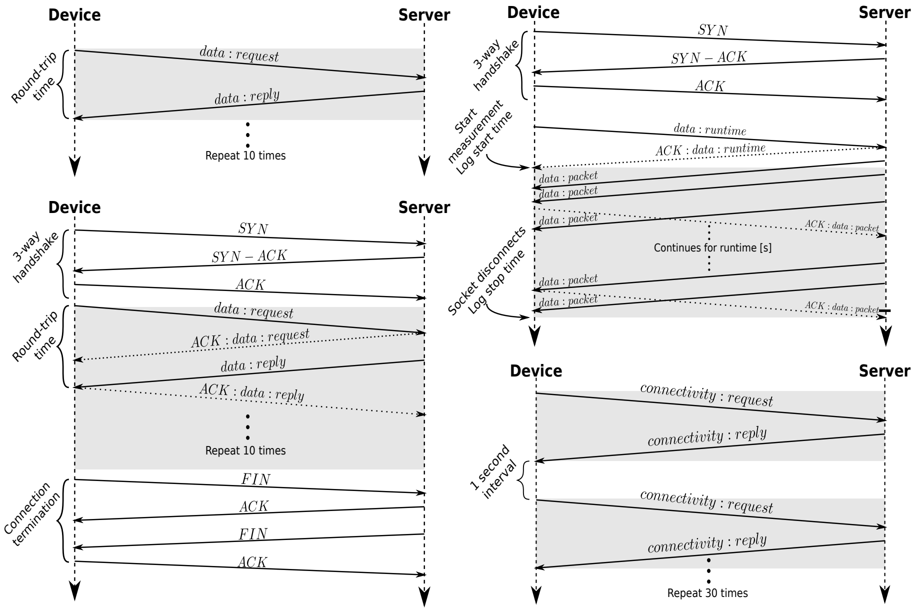


Fig. A.3: Top left: UDP RTT. **Bottom left:** TCP RTT. **Top right:** Throughput Downlink. **Bottom right:** Connectivity. Measurement flows are marked with grey areas.

performs. Based on these arguments throughput is considered the most important metric that is being measured in NetMap.

Round Trip Time (RTT): The RTT is, as defined by the IETF [11], the time it takes from a request is transmitted from the device to the server, and until the reply is received from the server at the device. The RTT is measured using both TCP and UDP, and only by measuring on payload packets, that is for TCP the connection is initiated before the measurements are performed. Both UDP and TCP RTT consist of 10 request/reply sequences, as illustrated in Figure A.3. When measuring the RTT it should be noted that it is not necessarily the same route that is being measured, i.e. the request to the server will travel by one route, while the reply might travel via another. Alternative delay estimations include one-way latency measurements, that would make it possible to compare the uplink and downlink of the connection, but requires synchronized clocks. The ICMP protocol is not used because it is not a transport protocol, why it is difficult to compare to other use cases. This choice is supported in [13] showing that TCP has an advantage in accuracy over ICMP.

The RTT is of interest because many applications operate using remote data, where most requests and actions performed by the user will cause a connection being made and data being transfered, why the delay induced by the network on the response time is of interest. UDP and TCP were chosen based on them being the most widely used protocols in Internet traffic.

Packet Loss or round-trip packet loss as defined by the IETF [11], is not measured using a dedicated method, but is extracted from the UDP RTT measurements, i.e. the number of packets not arriving within the timeout of 1 second set in the RTT measurements, effectively saving processing power and network traffic.

Connectivity: The connectivity is based on the IETF definition of the two-way interval temporal connectivity [11]. In the definition N request packets should be transmitted uniformly distributed between time T and time $T + \triangle T - W$, where W is the time to wait for a reply for each packet. For each sequence of N packets, if a single reply is received within the waiting time W the connectivity sequence returns true, and false otherwise.

The connectivity measurement in NetMap is only transmitting 1 UDP packet, then it waits for the reply with a timeout of 1 second, after which it waits 1 second and starts the sequence over, as illustrated in Figure A.3, which is done 30 times before the results are saved. In this way the connectivity requests are not randomly distributed, but they are not completely periodic either, because each request is spaced 1 second plus the varying RTT of the connectivity request/reply. Each request/reply will be evaluated to be either true or false, which is stored as the result in a vector indicating whether the individual request/reply sequence was a success or not. Furthermore, despite the connectivity running continuously, it is a cheap metric in terms of resources only using approximately 210 kB pr hour.

The connectivity is a discrete indication of the reachability of the server from the device at any point, while giving an indication of the continuous performance experienced. With the strict timeout defined for the metric, connectivity can be interpreted as an indicator of when the network performance drops below some defined threshold, which in this case means that it takes more than a second to do a request/reply sequence. The threshold, or timeout, of 1 second, is chosen as a response slower than this would be noticeable by an average user.

A.3.2 Physical Layer Connection Metrics

Network Context State Switch (NCSS): A NCSS is a change in state of the Cell ID (CID), Location Area Code (LAC), Network Type (Wifi or mobile connection), or Network Sub Type (3G, 4G, etc.) (see Figure A.1), and is measured, or rather registered, event based, why no data is transmitted in order to measure it. In Android the API PhoneStateListener [3] is used for this purpose.

It was chosen to register the NCSS because these are expected to impact the performance, e.g. in connection with a CID switch (in Figure A.1 when client #1 moves from CID 1 to 2) it is expected that the performance of RTT, throughput and connectivity will drop. By registering the CID and LAC switches the individual cell towers, and their coverage boundaries, can be identified and described in terms of performance as seen from the connected device. The performance of the tower is also expressed via the Network Sub Type, which indicates what connection technology is supported.

Received Signal Strength Indication (RSSI): RSSI is defined in [1] with values ranges from 0 to 31, which then can be translated into dBm, and is also registered via the Phone State Listener API in Android.

Registering the RSSI value of the connection currently being used by the device, helps identify why a connection might perform poorly. Adding the empirical RSSI measurements to the NPM, enables comparison to be made with the theoretical model based maps, provided by ISPs, to see if the experienced signal strength compare to the expected.

A.3.3 Notes Regarding NetMap Measurements

All the application layer QoS metrics are measured with active measurements, which is why they are performed sequentially, i.e. in a measurement sequence organized as [UDP RTT; TCP RTT; throughput uplink; throughput downlink], while connectivity is running continuously and the physical connection metrics are event based. The order in the measurement sequence is irrelevant but need to be non-overlapping to avoid one method influencing another.

When measuring public mobile networks the impact of the results can potentially rank an ISP as poorly performing, which emphasizes the importance of transparency of methods to evaluate causes and influencing factors on the results. Examples of influencing factors are non-dedicated devices, device quality, mobility of users, varying network load according to user activity, local environment conditions, and other.

In creating a NPM, it is vital that a sufficient number of measurements are used. But regardless of the number of measurements the NPM is based on, a number of indicators about the measurements in that area should be presented, e.g. number of measurements, distribution of measurements according to time of day or according to device type, and other.

To obtain a map, each collected measurement is tagged with a geographical point. Location information is preferably obtained using GPS achieving accuracy down to 4-5 meters. As having GPS receiver on is expensive in terms of power, alternatively other location estimators can be used, e.g. cell tower info, but with a much worse accuracy, possibly ranging from few hundred meters to more than 1 km. Other key metadata types logged are time, battery level, and manufacturer and device model.

A.4 Examples of Results

NetMap has not been rolled out in full scale to the public yet, but in an initial trial approximately 1500 users have participated in covering mainly the region of Aalborg and parts of the Northern Jutland region in Denmark, as a proof of concept. Additionally, we have designed and performed a number of tests gathering extensive measurements in a limited, selected in advance, geographical area. The amount of measurements collected in this way allows for statistical analysis. Selected results of these tests are presented below.

Indication of Network Load: Figure A.4 shows that the performance measured is influenced by the load on the network according to the time of day. The 2 vertical lines indicate the start and end of a typical work day, where the amount of people in the area and thereby the load on the network is higher, which is indicated by the measured throughput dropping in this timespan.

This shows that the measurements can be used to give a more precise forecast of what performance a user can expect from the network in a location at a given time. This can be utilized in a NPM but also in a prediction scenario where a network state forecast is used.

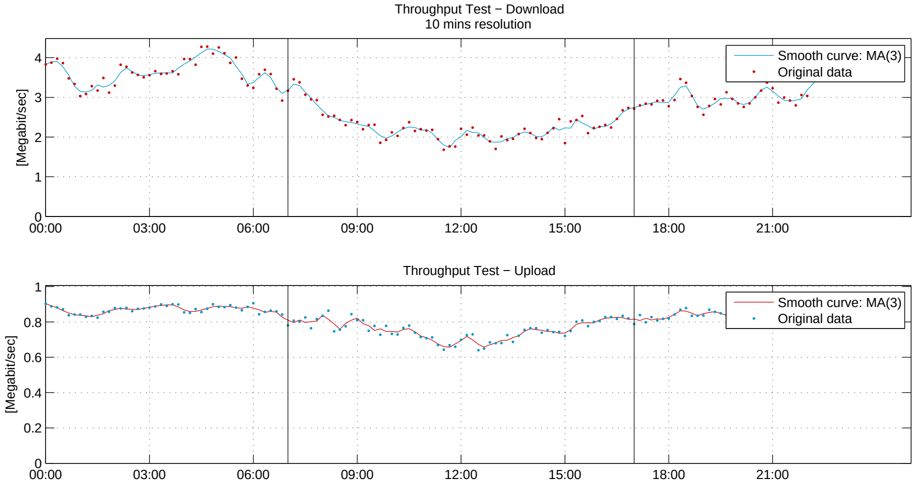


Fig. A.4: The throughput measurements are performed by 4 identical devices during 24 hours placed in the same location in the middle of campus, connected to the same ISP network, with the measurements being performed continuously with a few seconds of delay between each measurement sequence. The plot is based on 6175 throughput downlink and 6191 uplink measurements averaged over 10 min intervals to create datapoints, and the curve is smoothed with a MA(3) process.

Indication of Area Type Influence: Figure A.5 shows a tendency that the RTTs are lower in small town and rural areas than in urban areas. This is estimated to be due to a combination of the network resources available and the user density in the areas. This gives an indication of the balancing between network resources assigned by the ISP in the areas, and the users using the network.

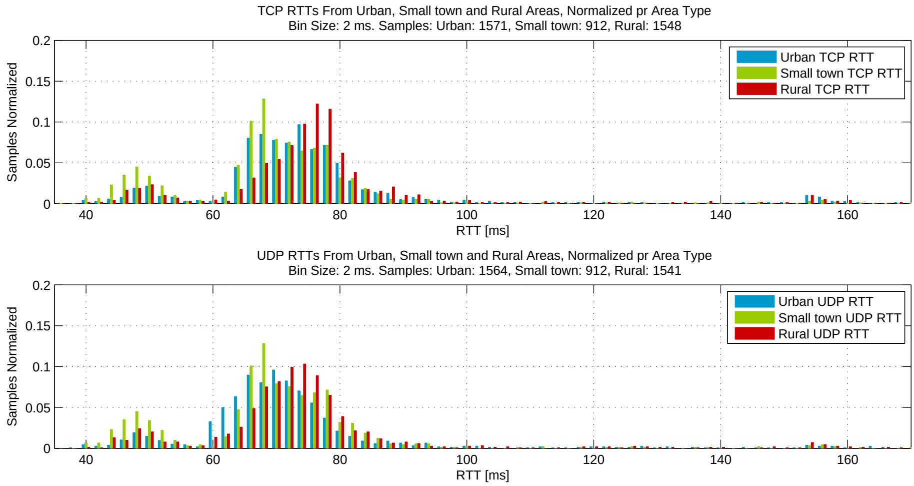


Fig. A.5: The RTT measurements are performed while driving in a car, following the speed limits, through different types of areas defined as urban, small town, and rural. The plot is based on 3689 TCP RTT and 3730 UDP RTT measurement sequences, where the means of the sequences are sorted in 2 ms bins.

Indication of Difference in Device Performance: Figure A.6 shows that 3 of the 4 devices used show the same performance tendencies as in Figure A.4, while the 4th device delivers such low performance that it is uninfluenced by the general network performance drop. This indicates that the device model can have a big impact on the performance measured, why it is important to include information about the device model in the metadata gathered along with the measurements.

A.5 Conclusion

In this paper a novel system called NetMap, for measuring network performance using crowd sourcing has been described, in which measurements are performed by users who have installed the NetMap App on their devices, and results are automatically submitted to Collection Point. The system measures network QoS metrics on the application layer, while available physical connection QoS metrics also are registered. Along with the measurements a

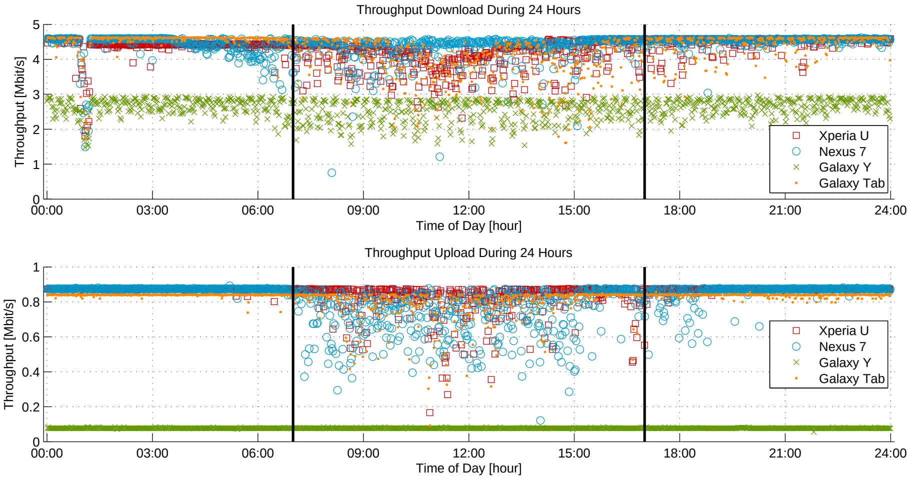


Fig. A.6: Same setup as described with Figure A.4, but with 4 different devices. The plot is based on approximately 1100 throughput uplink and 1100 downlink measurements pr device. Note that the sudden drop in downlink rates around 01:00 hrs is identified to be caused by backup being performed internally on the network where Measurement Point is located.

wide range of metadata including the state and context of the device while it is performing the measurements is gathered, to help in describing the measured performance and generating a NPM. The need for such a system, and such measurements, is based on the big increase in Internet traffic going through mobile networks, the increase in number of mobile devices, and that the existing network coverage maps are based on theoretical models rather than actual measurements. Finally some examples were given on the potential of the measurements obtained from NetMap, which includes indication of network load, balancing of network resources, and impact of the device on the measurements.

Outlook:

Further work to be done on NetMap involves the following points:

- Ensuring scalability of Measurement Point to profit from all active users.
- Optimized measurement methods to estimate QoS metrics.
- Developing a more intelligent scheme for scheduling measurements, i.e. to accommodate if more measurements are needed in one area than another.
- Extrapolation of measurements in NPM to cover areas where few measurements are performed.

- Additional motivation of users for using NetMap, e.g. including competitions, sponsors and rewards.

References

[1] 3GPP, *TS 27.007 Specification Detail*, 2014, ch. 8.5 Signal quality +CSQ, http://www.3gpp.org/ftp/Specs/html-info/27007.htm.

[2] 4G Americas, "3g/4g deployment status," May 2014. [Online]. Available: http://www.4gamericas.org/index.cfm?fuseaction=page&pageid=939

[3] Android Developer, "Phone state listener," May 2014, http://developer.android.com/reference/android/telephony/PhoneStateListener.html.

[4] Cisco, "Cisco visual networking index: Forecast and methodology, 2012–2017," May 2014. [Online]. Available: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/ip-ngn-ip-next-generation-network/white_paper_c11-481360.html

[5] DeIC, "Danish research network," May 2014, http://www.deic.dk/en/node/171.

[6] Erhvervsstyrelsen, Denmark, "Mobil datatrafik fordoblet paa et aar (translate: Mobile data traffic doubled in a year)," April 2014. [Online]. Available: http://erhvervsstyrelsen.dk/pressesoeg/694899/5

[7] J. Mankowitz and A. Paverd, "Mobile device-based cellular network coverage analysis using crowd sourcing," in *EUROCON - International Conference on Computer as a Tool (EUROCON), 2011 IEEE*, April 2011, pp. 1–6.

[8] NetMap Measurement System, "Netmap measurement system wiki," May 2014, http://sourceforge.net/p/netmap-system/wiki/Home/.

[9] Open Signal, "Open signal," May 2014, http://opensignal.com/.

[10] Telia Denmark, "Coverage map," May 2014. [Online]. Available: http://telia.dk/mobil/kundeservice/daekning

[11] The Internet Engineering Task Force, "Ip performance metrics," May 2014, http://datatracker.ietf.org/wg/ippm/charter/.

[12] Unitz LLC, "Down right now," May 2014. [Online]. Available: http://downrightnow.com/

[13] L. Wenwei, Z. Dafang, Y. Jinmin, and X. Gaogang, "On evaluating the differences of {TCP} and {ICMP} in network measurement," *Computer Communications*, vol. 30, no. 2, pp. 428 – 439, 2007. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0140366406003719

Paper B

Performance Evaluation of Methods for Estimating Achievable Throughput on Cellular Connections

Lars M. Mikkelsen, Nikolaj B. Højholt, and Tatiana K. Madsen

The paper has been published in the
Proceedings of 15th International Conference, NEW2AN, and 8th Conference, ruSMART pp. 422–435, 2015.

Abstract

The continuous increase in always connected devices and the advance in capabilities of networks and services offered via the network is evident. The target group of most of these devices and services is users, so how users perceive the network performance is of great importance. Estimating achievable throughput (AT) is the main focus of this paper, which can be expressed as the data rate that users experience. We establish the Bulk Transfer Capacity (BTC) method as the ground truth of the AT. We choose to evaluate the Trains of Packet-Pair (TOPP) method as an alternative to BTC in estimating AT, due to its much reduced resource consumption. Based on real-life measurements of the two methods we conclude that TOPP is a good candidate to estimate AT, based on similarity in results with BTC.

B.1 Introduction

The usage of mobile data has increased dramatically in recent years and it will continue to increase in the coming years [1]. This is due to the introduction of always connected smart devices, due to the capabilities of the networks, and due to service types being offered to users. A result is users getting used to having services available everywhere and all the time via networks. For this reason it is critical that the networks deliver continuously high quality service and only rarely drop in performance. Generally, network behavior can be captured via network monitoring, and the information about the connection quality that the network provides can be obtained through network performance measurements, performed actively from devices connected to the network.

When discussing network performance one of the important things is estimating the data rate that can be achieved for connected devices. In measuring data rates on networks it is important to be clear on what the goal is, i.e. what you want to measure, and how this is measured. Overall there are 3 different types of results related to data rates: capacity, available bandwidth (AB), and achievable throughput (AT).

- Capacity, often called minimum capacity, is a measure of the overall capacity of the link.
- AB is a measure of the unused capacity of a link. As there typically is traffic from many users on a link, AB is the upper bound for data rate of a transmission on the link for a user.
- AT is a measure of what portion of the link that the user actually is assigned when performing a transmission. So this can be seen as a measure of what is experienced by a user.

Our goal is to find a good method to deploy in measurement systems aimed at providing end users with information about the connection quality they get. From the perspective of a user it is not that relevant to know what the capacity of the connection is, because he will most likely never experience this. The same argument could be applied to AB as it is an upper bound of what can be experienced. AB could potentially be much higher than what the user will ever experience. For this reason the focus of this paper will be on estimating AT as experienced from the end user.

We consider a case of mobile cellular networks. It is known that bandwidth estimation is challenging for this type of networks [2], and only very limited attempts exist to adapt measurement techniques for this use case [4].

Estimating data rates is a field where a great amount of research has been done, and where many different approaches exists [9]. Traditionally most approaches aim to either estimate capacity or AB for wired networks with first-come first-served store-and-forward links. Recently estimations of AT started to get some attention, as this metric is conceptually different from AB for cellular links. The majority of AB estimation methods are developed for traditional LAN and WLAN networks and not specifically for mobile networks. But as user devices typically are connected to the Internet via mobile networks, this is a big concern. Mobile networks are managed much more dynamically than LAN and WLAN networks due to high mobility of the connected devices. This should be taken into account when selecting and measuring the network performance. So in selecting the method to use, one has to consider the network type, how connected devices act, and how the network is managed.

Another issue in estimating AT on mobile networks is that the bandwidth assigned to a user varies much more rapidly than for other network types. Consequently the challenge is to estimate the throughput instantaneously, to counter the rapid changes in connection performance. Additionally when doing active measurements the goal is to reduce overhead and put as little traffic on the network as possible. But when doing estimations fast, the amount of data on which to base the estimate is reduced, making the estimate less confident. So the trade-off is the time spent and the amount of data collected during a measurement versus the correctness of the result.

In practice there is a great need for AT estimation methods aimed at mobile networks from tools such as Open Signal, NETRADAR, and NetMap.

Open Signal [8] uses crowd sourcing to perform signal strength measurements used to generate a network coverage map, and to localize base stations. Furthermore, Open Signal also allows the user to perform individual measurements such as round trip times (RTT) and throughput estimation.

NETRADAR [7] also applies crowd sourcing to gather signal strength measurements and perform network performance measurements such as RTT and AT. The main focus of this system is to analyze the correlation between signal strength measurements and network performance measurements.

In NetMap [6] crowd sourcing is used to performance measurements such as RTT and throughput estimation. The goal of this system is to make a network performance map (NPM) that should replace the network performance maps provided by ISPs. The NPM provided from ISPs are based on signal strength measurements and theoretical models of these, and from this the network performance metrics such as RTT and throughput are estimated.

In the present work it is chosen to look at active measurement methods that attempt to estimate the AT. Within this area two methods have been chosen that are very different in resource consumption; Bulk Transfer Capacity (BTC) which is a traditional method that applies a very simplistic brute force approach in estimating the AT; and Trains Of Packet Pairs (TOPP) that applies a more delicate approach utilizing information of packet interarrival times in estimating the AT. These two methods will be compared based on the results from a number of measurements. The measurements will be done on a commercial mobile network in Denmark with devices connected via 4G (LTE).

The rest of the paper is organized as follows: Section B.2 will describe what we will measure and the concept of ground truth used in this paper. Section B.3, B.4 and B.5 will describe measurement setup and the measurement methods used. In Section B.6 results from the method will be compared and evaluated. Lastly in Section B.7 the results will be discussed and conclusions will be drawn.

B.2 What Will Be Measured

When estimating anything it is important to know if your estimation is close to the actual value of what is being estimated. The actual value can be denoted as the ground truth - the true value of what is being estimated. In our case we are estimating the AT, i.e. how much data the user is allowed to put through his connection. The challenge occurs when we are performing estimations based on real-life measurements. The question is then which of the measured values is the ground truth. That is if any of the obtained values is the ground truth. This problem could be overcome by using simulations instead. In simulations it is possible to extract all information about users and network states. This information is often not available when making real-life measurements, and in many cases they are not observable.

There will be fluctuations of the AT for the user due to different factors. These factors include cross-traffic (load sharing among multiple users), channel conditions and interference, and mobility and geographical location of users. Only the first aspect related to cross-traffic is present in wired networks, while all the other aspects are specific to wireless access networks.

Additionally, cellular networks employ a proportional fair scheduler that allocates resources to users, taking into account change in channel conditions and interference, e.g. providing some compensation for users near the cell edge.

The fluctuations of AT can be roughly divided into fast and slow. The slow fluctuations will be caused by the general user load of the network changing according to location and over time. The fast fluctuations are related to when other users connect to or leave the network, and when other users start or stop transmitting on the network. Additionally, under loaded network conditions a user with bad signal conditions will not be scheduled until the scheduler predicts that the user will experience good signal to noise ratios. On cellular networks resource allocation to the users are done every transmission time interval, which is as low as 2 ms for HSPA and 1 ms for LTE. It is not feasible to measure with such high resolution, and the measurements would be required to be done on the physical layer. Furthermore, as we are interested in the user experience, which also relates to higher layers, we will measure no lower than the transport layer. On the transport layer the measurement frequency is however limited by the scheduler of the device, and by the chosen size of the samples. Despite these limitations we will still be able to observe fluctuations in the throughput, only less rapid than what potentially could be observed on the physical layer.

As AT is a measure of how much data the user actually is allowed to put through a connection, we will approximate the ground truth using the method Bulk Transfer Capacity (BTC). BTC is measured by putting as much data as possible through a connection over a duration of time, using TCP. This represents a typical user file transfer, as TCP is the most widely used transport protocol on the Internet. Furthermore, by using TCP, which employ flow control and congestion avoidance, the data rate is controlled and adjusted according to the capabilities of the user connection, just as any other file transfer would be.

B.3 Measurement Setup and Methodology

Implementation:

BTC (see Section B.4) is implemented using TCP and TOPP (see Section B.5) implemented using UDP. The client implementation is done as an Android application in Java, and the server implementation is also done in Java. For this reason the samples are done on the application layer, why packet size is containing payload + IP header + UDP/TCP header. The measurements are performed between a mobile device and a server acting as a measurement point for all the measurements.

Client Device:

All the measurements are performed with the same Android device (Samsung Galaxy S5 (SM-G900F)) running Android 5.0 (Samsung stock Lollipop). The device is connected to the Internet using a LTE (4G) connection to a large danish Internet Service Provider (ISP). All measurements are done while being stationary at the same location near (<500m) the cell tower. The location of the client device is indoor at the Aalborg University campus.

Measurement Server:

The measurement server is placed at Aalborg University and is connected to the Internet using a Gbit connection to the danish research network, making the shortest path for data from the device to the server through the ISP network, via the Danish Internet Exchange point onto the danish research network, to the server. The server is a stationary PC running Ubuntu 14.04, where no other software than the measurement software is running during the test.

B.4 Bulk Transfer Capacity (BTC)

The BTC method, as defined by the Internet Engineering Task Force (IETF) [10], is the average amount of data that can be transmitted through a link per second, $BTC = data\_sent/elapsed\_time$. Here $data\_sent$ is the amount of unique bits transmitted excluding headers. $elapsed\_time$ is the time between receiving the first bit and the last. A way to implement BTC is to produce samples and to stop the measurements when a predefined number of samples is received, or when a certain time to do measurements has elapsed. The size of a sample is fixed pr measurement and the value for the sample size should be selected in a clever way, e.g. adjusting it according to the currently used connection technology. This will help to achieve more comparable results and additionally to some extend limit the amount of data used per test. Figure B.1 shows a sequence diagram of BTC for the downlink case.

BTC is implemented using TCP which applies flow control mechanisms in terms of slow start and congestion avoidance that throttles the data flow according to the capabilities of the connection. This causes the samples of the BTC measurement to fluctuate a lot, as TCP attempts to adjust the rate according to the capabilities of the connection. This can be seen from Figure B.2.

From Figure B.2 it can be seen that there is a big spread on the individual samples, why some processing must be performed to get an idea of the tendency of the samples. This is done by calculating a mean of the samples, which is the red line on the plots. The mean is calculated for sample 1 to

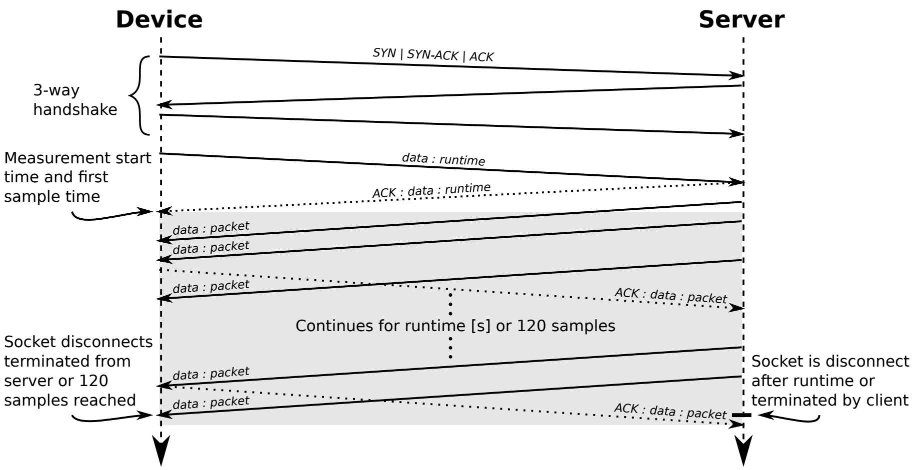


Fig. B.1: Sequence diagram of a BTC download measurement.

sample n, where n runs from 1 to the total number of samples. This reveals the impact that the flow control of TCP has on the data transfer, in that the mean rises in the beginning and settles after approximately 3 seconds.

The slow start should be taken into account when performing the measurements, or rather when processing the results. One approach could be to only calculate the mean of samples after the initial 3 seconds, and discard the initial samples. However, as TCP is chosen as the transport protocol, the rise time of the throughput is an integrate part of the result, as it says something about the stability of the connection.

Another thing to consider is for how long to measure, or alternatively how much data to transmit. With a long measurement the result will be more stable, but at the same time more expensive in terms of data usage. And conversely, with a short measurement the result will be more affected by fluctuations, but it will be cheaper in data usage.

B.4.1 BTC Setup

In our implementation when performing a BTC measurement the bytes are counted at the receiver side and sampled. We have selected the following parameters: Each measurement consists at most of 120 samples, or a duration of 15 sec. Thus, a single measurement round might include less than 120 samples if the connection is slow at the time of measurement (e.g. below 1 Mbit/s). The time of receiving each sample is stored for use in the processing phase. The size of a sample is adjusted according to a preliminary measurement, which allows us to roughly control the duration of the measurement while getting the same amount of samples for processing. In the preliminary

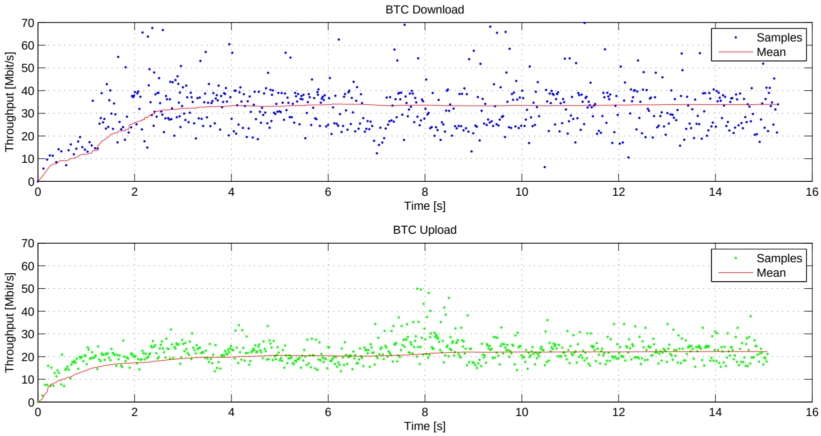


Fig. B.2: Example of samples obtained via a BTC measurement on a LTE connection.

measurement a sample is $30 \cdot 5000$ bytes, only considering payload bytes. Header bytes are added to the calculation during processing of the results. After the preliminary measurement the size of a sample is adjusted such that a measurement approximately takes 10 seconds. For each measurement for simplicity a single result will be obtained as the mean of all samples.

B.4.2 BTC Evaluation

To evaluate this setup 20 BTC measurements are performed both for download and upload on a LTE connection. The results are represented in Figure B.3.

From Figure B.3 it can be seen that the means of the measurements, both for upload and download, are rather steady around the same value.

Furthermore, from Table B.1 it can be seen that the data usage is quite extensive. This is due to the size of sample being fixed such that each measurement approximately lasts for 10 seconds, to ensure that the rate will have time to settle.

	Mean Throughput [Mbit/s]	Variance of Means [Mbit/s]	Mean Data Usage [MB]	Mean Time [s]
Download	39.9049	13.5181	55.1067	12.91
Upload	10.8834	2.3749	8.3098	7.07

Table B.1: Statistics based on 20 BTC download and upload measurements.

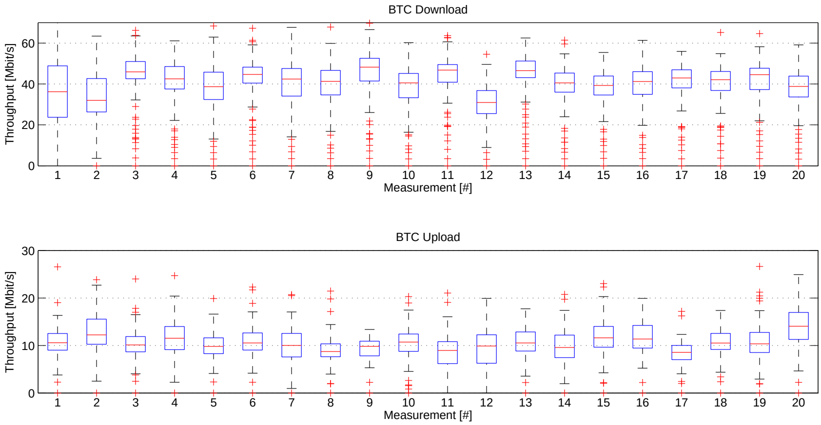


Fig. B.3: Boxplot of 20 BTC download and upload measurements. The box corresponds to the center samples between the 25th and 75th percentile, the red line in the box is the mean, the whiskers extend to the most extreme points that are not considered outliers, and the red pluses are outliers. Note: download plot is limited to 70 Mbit/s and upload to 30 Mbit/s.

B.5 Trains Of Packet-Pair (TOPP)

B.5.1 BTC Alternatives

Based on the results in Section B.4.2, it is apparent that the resource usage of BTC is very high. So another method is needed that preferably achieves similar results, but significantly reduces the resource consumption both regarding data and time. There exists a wide range of methods developed for estimating AB. These methods generally fall into one of two categories: packet rate methods and packet gap methods [4]. Packet gap methods pose high requirements to measurement equipment and for that reason is a bit more tricky to implement in our setup. Our setup consist of standard mobile phones, why the necessary time resolution of measurements is not supported. Therefore we choose a method from the packet rate method class, namely Trains of Packet-Pairs (TOPP). TOPP is a well known technique for estimating AB, and some studies have indicated that using AB techniques for measuring in cellular environments would result in AT estimations [4]. This motivates our choice and calls for more extensive studies of its applicability for AT estimation.

B.5.2 TOPP Description

In order to describe TOPP [5] the principle of packet-pair will be described first. The idea behind packet-pair is sending two packets back-to-back with the same payload L. When the packets arrive at the receiver the packets will arrive with a dispersion δ as illustrated in Figure B.4.

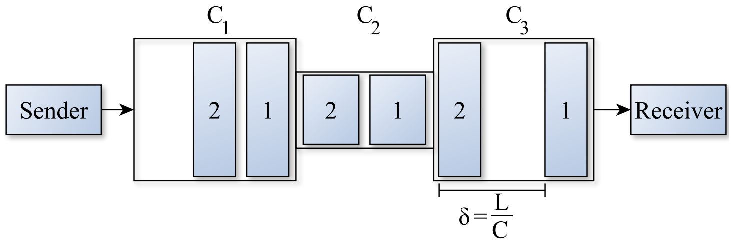


Fig. B.4: The height of C_1, C_2 and C_3 simply illustrates the capacity of the link between sender and receiver. The packets leave the sender back to back, and they arrive at the receiver with a dispersion that is determined by the narrow link.

The UDP protocol is used as it does not implement flow control. The dispersion δ is given by the payload L over the capacity of the link C. From this it can easily be deduced that $C = L/\delta$. As proposed by [3] the larger the payload L the greater the impact the cross traffic will have on the measurements. Therefore, for big L the measurements are trending towards the tight link instead of the narrow link. Whereas if L is small the result trends towards the narrow link capacity or post-narrow link capacity.

TOPP is based on the packet-pair principle with an increase in the number of packets transmitted ($N > 2$). The packets are sent to a receiver with a fixed packet size. When the number of packets N is sufficiently large the dispersion δ between all packets becomes an average packet delay. The total dispersion $\Delta(N)$ can be found by:

$$\Delta(N) = \sum_{n=1}^{N-1} \delta_n \tag{B.1}$$

The measured AT can thereby be calculated by:

$$A = \frac{(N-1) \cdot L}{\Delta(N)} \tag{B.2}$$

B.5.3 TOPP Setup

When performing a TOPP measurement the transmitted packets are counted at the receiver side, and the time of arrival of each packet is noted and used during the processing phase. For an initial test of the method we have choosen the following parameters: Each measurement consists of a maximum of 800 packets. The amount of packets counted at the receiving end can potentially be lower due to packet loss during the measurement. The packet size is set to 1500 bytes, based on the standard MTU size. However, we are aware that due to fragmentation and packetization done by link layer protocols in cellular networks, chunks of data of 1500 bytes will end up being transmitted in different physical layer packets, and thus, other packet sizes might be preferable.

Furthermore, it was discovered that for the download case there was a high amount of packet loss, while none for the upload case. We believe this is caused by the server having a network connection with high capacity (1Gbit/s), while the client connection capacity is significantly lower (LTE Cat4 theoretically 150 Mbit/s downlink, 50 Mbit/s uplink). For this reason we choose to limit the download transmit rate at the server to 200 Mbit/s, which still leaves room for packets to be transmitted faster than what the device can handle. This approach significantly reduces the packetloss for the download case.

B.5.4 TOPP Evaluation

In Figure B.5 the results of 20 TOPP measurements performed on a LTE connection can be seen.

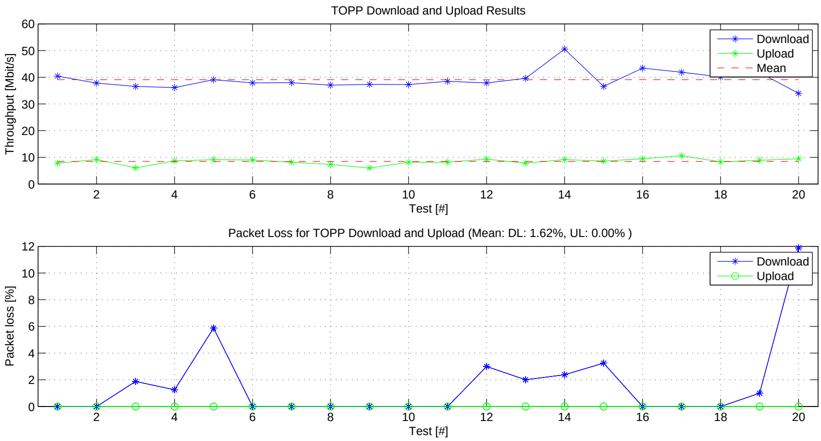


Fig. B.5: Top plot: The results of 20 TOPP download and upload measurements performed on a LTE connection, and the mean of the download and upload results respectively. Bottom plot: The packet loss registered in the TOPP measurements.

From Figure B.5 it can be seen that both for download and upload the results are mostly very consistent, only fluctuating very little. Furthermore it can be seen that there still is some packet loss on the download measurements, however much less than what previously was the case, and the packet loss that is present does not seem to have any impact on the results.

To be able to compare TOPP with BTC the statistics of these TOPP measurements are presented in Table B.2. From this table it can be seen that the results in terms of mean values and variance of means are similar. The variance for the TOPP means is slightly lower, than for BTC. TOPP only uses approximately 2% and 14% of the data that BTC uses for download and upload respectively. Another thing to notice is that TOPP uses significantly less time pr measurement, why the 20 measurements can be performed much quicker than BTC, leaving less time for the network to change.

	Mean Throughput [Mbit/s]	Variance of Means [Mbit/s]	Mean Data Usage [MB]	Mean Time [s]
Download	39.0834	12.2029	1.1444	0.23
Upload	8.4744	1.2309	1.1444	1.09

Table B.2: Statistics based on 20 TOPP download and upload measurements.

B.5.5 Considerations

When comparing TOPP to BTC, they apparently apply two significantly different methods. In BTC data is transmitted as fast as possible, where the rate is throttled by TCP. At the receiver side the input buffer is then read, independent of individual packets. Because of this the transmission of data can be seen as a stream of data that is sampled during the measurement. In TOPP the data is also transmitted as fast as possible, but here the sampling is being done on a pr packet payload basis. Sampling based on the packets, instead based on the overall data stream, means that TOPP will be much more prone to be influenced by fast fluctuations in the network performance. So when the packet size and the number of packets in TOPP is increased, TOPP begins to look more and more like BTC.

An important thing to consider is the parameters that is defined for TOPP, as they may have a big impact on the performance of TOPP in terms of achieving a result as close to the ground truth as possible. As it was just discussed, when the number of packets and the packet size is increased, TOPP goes toward BTC in concept. So this is a trade off as the goal is to have an accurate method for estimating AT, that consumes as little data and time as possible.

B.6 Comparison of BTC and TOPP

To evaluate whether TOPP is sufficiently accurate in estimating the AT, a number of tests will be done where the results will be compared to the ground truth that will be obtained using BTC. In the comparison the parameters of TOPP will be changed to find the best trade off between data usage and accuracy of the results. The different combinations of TOPP settings will furthermore be evaluated in different scenarios. The network load will be different in the different scenarios, so it can be examined what impact this has on the results of TOPP.

B.6.1 Measurement Setup

In Table B.3 the different combinations of TOPP settings can be seen. 3 different number of packets and 3 different packet sizes have been chosen. For each set of 3 TOPP measurements with the same number of packets, a BTC measurement will be performed, which will serve as the ground truth for that set of TOPP measurements. The BTC measurements will be performed as described in Section B.4. For each unique setup of TOPP, and for each BTC,

	300 bytes	1500 bytes	3000 bytes
200 packets	a1	a2	a3
800 packets	b1	b2	b3
1500 packets	c1	c2	c3

Table B.3: TOPP setttings matrix.

the measurement will be repeated 20 times. Based on this, a full schedule of measurements is as follows:
BTC; TOPPa1; TOPPa2; TOPPa3; BTC; TOPPb1; TOPPb2; TOPPb3; BTC; TOPPc1; TOPPc2; TOPPc3.

A full schedule of the measurements covering all settings are done both at noon and at night, but at the same location as described in Section B.3. This will allow for evaluation of how the setups are influenced by cross traffic on the network, as during night there are only very few users in the area.

B.6.2 Results Evaluation

In Figures B.6 and B.7 the results are presented for download and upload measurements respectively.

From Figure B.6 it can be seen that the download rates measured at noon are slightly higher than during night. Furthermore, for most of the cases the night results seem to be less fluctuating than the day rates, indicated by slightly smaller confidence intervals. For TOPP the best setup, i.e. the results closest to the BTC results, seem to be with the bigger packet sizes (1500 and 3000 bytes). The number of packets pr measurement does not seem to have any impact on the mean, but there is a slight tendency that the confidence intervals are lower for bigger number of packets.

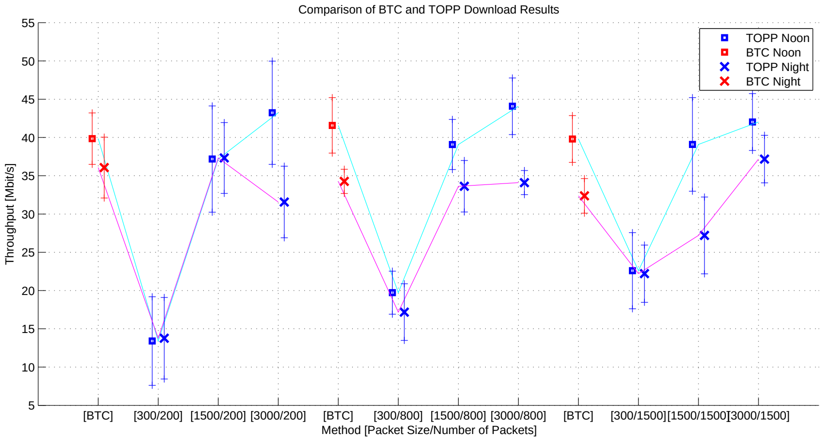


Fig. B.6: Comparison of BTC and TOPP download measurements performed at noon and at night. Each result is based on 20 measurements where the square or x marks the mean. The whiskers indicate the 95% confidence interval of the means. Whether the result is for BTC or TOPP with a certain setup is indicated in the x-axis label. From left to right each BTC result is linked with the following 3 TOPP results, according to time of day.

From Figure B.7 it can be seen that the upload rates almost consistently are higher at night than at noon. Also here the bigger packet sizes for TOPP seem to achieve the results closest to BTC. But the confidence intervals for upload seem to be slightly smaller at noon than at night.

Generally the confidence intervals for TOPP are neither consistently bigger or smaller than for BTC. Finally it can be noted that the packet loss during the TOPP measurements is highest when the packet size is small, independent of the number of packets. However, for the highest number of packets the biggest packet size also shows high packet loss.

B.7 Discussion and Conclusion

Based on the results from the two measurement methods, it can be concluded that TOPP is a good candidate for estimating AT. What is interesting about this result is that the measurements are done in a real life network, where the cross-traffic and varying user load is very real. From the evaluation of the results it seems as TOPP offers a good trade-off between delivering results similar to BTC and not consuming a lot of resources. The similarity in the results both relates to the means of the measurements being close to the means of BTC. But also the fluctuations in the means are similar to those of BTC. Furthermore, TOPP seems to deliver good results both in the heavily and the lightly loaded network scenario.

From the presented results the recommendation for the settings of TOPP

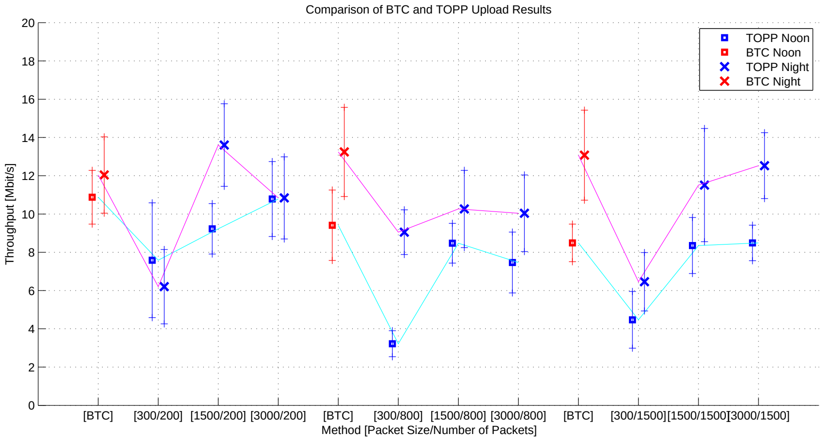


Fig. B.7: Comparison of BTC and TOPP upload measurements performed at noon and at night. See Figure B.6 for description.

would be a packet size 1500 bytes, and 800 packets pr measurement. However, further measurements and evaluations based on more packet sizes and different number of packets could reveal setups that consume even less resources while still being stable in results. Also, to better be able to compare the methods it should be attempted to perform the measurements simultaneously on several devices. Other future work should include a deeper analysis of the impact of the cellular network scheduler on the measurements should be performed. Furthermore, there should be performed further evaluation of the validity of BTC as provider of the ground truth. And lastly, other alternative measurement methods should be tested and evaluated along with TOPP.

The results presented in this work are obtained in a very limited scenario. The scenario could be expanded both measuring at more times and at more locations relative to the base station. This would yield stronger results and a better general image of the performance of the methods. Furthermore, to get a complete understanding of the dynamics in cellular networks, the measurements should also include cellular specific parameters such as signal strength and parameters related to mobility.

References

[1] Cisco, "Cisco visual networking index: Forecast and methodology, 2012–2017," May 2014. [Online]. Available: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/ip-ngn-ip-next-generation-network/white_paper_c11-481360.html

[2] U. Devi, H. Viswanathan, R. Kokku, V. Pichapati, and S. Kalyanaraman, "On the estimation of available bandwidth in broadband cellular networks," in *Sensing, Communication, and Networking (SECON), 2014 Eleventh Annual IEEE International Conference on*, June 2014, pp. 19–27.

[3] C. Dovrolis, P. Ramanathan, and D. Moore, "Packet-dispersion techniques and a capacity-estimation methodology," *Networking, IEEE/ACM Transactions on*, vol. 12, no. 6, pp. 963–977, Dec 2004.

[4] D. Koutsonikolas and Y. Hu, "On the feasibility of bandwidth estimation in wireless access networks," *Wireless Networks*, vol. 17, no. 6, pp. 1561–1580, 2011. [Online]. Available: http://dx.doi.org/10.1007/s11276-011-0364-5

[5] B. Melander, M. Bjorkman, and P. Gunningberg, "A new end-to-end probing and analysis method for estimating bandwidth bottlenecks," in *Global Telecommunications Conference, 2000. GLOBECOM '00. IEEE*, vol. 1, 2000, pp. 415–420 vol.1.

[6] L. M. Mikkelsen, S. R. Thomsen, M. S. Pedersen, and T. K. Madsen, "Netmap - creating a map of application layer qos metrics of mobile networks using crowd sourcing," in *Internet of Things, Smart Spaces, and Next Generation Networks and Systems*, ser. Lecture Notes in Computer Science. Springer International Publishing, 2014, vol. 8638.

[7] NETRADAR, "Netradar," September 2014, https://www.netradar.org/en.

[8] Open Signal, "Open signal," May 2014, http://opensignal.com/.

[9] R. Prasad, C. Dovrolis, M. Murray, and K. Claffy, "Bandwidth estimation: metrics, measurement techniques, and tools," *Network, IEEE*, vol. 17, no. 6, pp. 27–35, Nov 2003.

[10] The Internet Engineering Task Force, "Ip performance metrics," May 2014, http://datatracker.ietf.org/wg/ippm/charter/.

Paper C

Verification of 3G and 4G Received Power Measurements in a Crowdsourcing Android App

Mads Lauridsen, Ignacio Rodriguez, Lars Møller Mikkelsen,
Lucas Chavarria Gimenez, Preben Mogensen

The paper has been published in the
Conference of IEEE Wireless Communications and Networking, 2016.

Abstract

Many crowdsourcing Android applications are available for measuring network Key Performance Indicators such as received power, latency, and throughput. The data is useful for end-users, researchers, and Mobile Network Operators, but unfortunately the applications' accuracy are rarely verified.

In this paper we verify the crowdsourcing Android application NetMap's ability to measure LTE Reference Signal Received Power by analyzing the Root Mean Squared Error, being 2-3 dB, and cross-correlation coefficient, being above 0.8, with measurements obtained by use of a professional radio network scanner and measurement phones. In addition, the application is applicable, but less accurate, for 3G Received Signal Code Power measurements. The studies are made for various device speeds and in different scenarios including indoor, urban, and highway, where the NetMap application is showed to perform well.

C.1 Introduction

Obtaining Key Performance Indicators (KPIs), such as received power, latency, throughput, and mobility performance for mobile networks is of interest to the end-user, researchers, and Mobile Network Operators (MNOs) [4]. The end-user can use the KPIs when selecting MNO subscription, while access to the KPI data enables researchers to study network problems and develop potential solutions. Finally the KPIs can assist MNOs in optimizing their network deployment and setup.

The KPIs can be measured using drive tests, dedicated test beds, network-side-only tools, or user-deployed applications, [4]. The first 3 solutions often rely on professional tools, only cover a limited area, and require many man-hours of work to be conducted. On the contrary the user-deployed applications enable both the end-user, researchers, and MNOs to obtain the KPIs, reflecting the real end-users experience and mobility, at a low cost. Many applications, e.g. [6, 8, 10–12], have started using crowdsourcing i.e. spreading the applications among many users to gather as much data as possible.

The aforementioned applications are able to measure a large number of parameters including received power, latency, throughput, location, mobility performance, and energy consumption. In addition, they are able to cover a larger geographical area as compared to what drive tests and dedicated test beds can, but unfortunately the developers rarely verify whether the measurements are accurate. In [8] the authors study how accurate the latency and energy consumption measurements are, while [10, 12] compare what they termed "manual measurements" and subsets of their own data without giving further details. In [6] the authors focus on the energy consumption of running the application, which is of high importance as crowdsourcing will be difficult if the application has a reputation of excessive energy consumption. The quality of the received power measurements is discussed in [11] which observed that measurements are averaged by the phone and that some phones seem to report inaccurate numbers. Related to that, the authors of [3] state that it is likely that different phones report with different level of resolution, but the authors don't examine it in further detail.

The received power is important for understanding other network KPIs such as latency and throughput, because it will affect the applied modulation and coding scheme, and the number of retransmissions. However, according to the survey in [4] only 14 of 29 surveyed tools are able to produce coverage maps or report the received power. In fact, the conclusion of [4] specifically mentions that the accuracy of the tools is difficult to compare. This entails a root cause analysis of the observed network KPIs may be difficult to perform.

The contribution of this paper is to verify the received power measurement accuracy of our crowdsourcing Android application, named NetMap, which uses the Android API [5]. Having verified and accurate received power measurements enables researchers and MNOs to understand other KPIs such as latency and throughput in further detail. We perform the verification by comparing the NetMap measurements with 2 professional measurement phones and a radio network scanner in 4 different scenarios including indoor, urban, and highway at speeds from pedestrian to 110 km/h.

The paper is structured as follows; first the NetMap application is described in Sec. C.2 with focus on how received power measurements are made, then the verification methodology including tools, scenarios, data processing, and evaluation is presented in Sec. C.3. Selected results are presented together with a discussion and future use of the application in Sec. C.4 and C.5 respectively, followed by the conclusion in Sec. C.6.

C.2 The NetMap Application

The NetMap Android application is designed to capture network performance on the application layer and Radio Access Technology (RAT) specific parameters that affect the end-user experience [7]. The application captures information for 2G, 3G, and 4G while also logging user position via GPS.

The default NetMap application logs throughput, connectivity, network context state, Round Trip Time (RTT) and received power [7]. In this work, a simplified version logging the two latter parameters was applied. Both the RTT and received power measurement are collected with a frequency of approximately 1 Hz, which entails the mobile terminal is always expected to be actively connected to the serving cell. In between the measurements the application is in a sleep state to reduce the effect on battery life and general

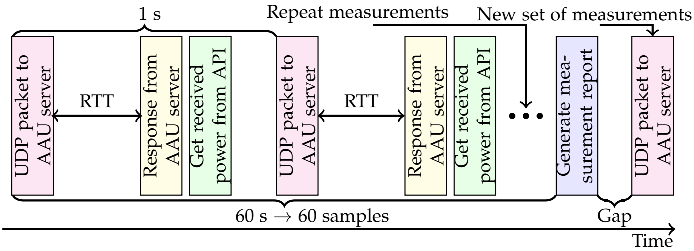


Fig. C.1: NetMap activity chart.

resource usage.

The RTT measurement is initiated when the application sends a UDP packet, with a payload containing a packet ID of a maximum of 1024 bytes, to a server at Aalborg University (AAU). When the response is received at the application layer the total RTT is logged. This is implemented using the DatagramPacket and the DatagramSocket APIs in Android [5]. A timeout of 1 s is set on the socket to capture long RTTs.

Every time a RTT measurement is performed the received power is also sampled as illustrated in Fig. C.1. When 60 measurements are completed a report is generated after which 60 new measurements are initiated as soon as possible. Depending on the RAT it varies how the received power is calculated and which Application Programming Interface (API) must be used. Furthermore, depending on the state of the phone's screen the APIs act differently, and in addition, different APIs are available in different versions of the Android operating system.

There are two APIs that can be used for extracting the received power values; SignalStrength and CellInfo [5]. The SignalStrength API has been available since Android SDK version 7 (Android 2.1). This API offers a wide range of received power related information, and for 3G measurements NetMap reads the GsmSignalStrength via the call getGsmSignalStrength(). The 3G received power is reported via this API because Android decided it is convenient that the received power for different RATs is reported in the same place. The call returns an Arbitrary Strength Unit (ASU) value representing the 3G Common Pilot Channel Received Signal Code Power (RSCP), and valid values are (0-31, 99) as defined in [1]. The conversion to RSCP is defined as:

$$\mathrm{RSCP} = \mathrm{ASU} - 120 \qquad [\mathrm{dBm}] \tag{C.1}$$

For Long Term Evoloution (LTE) the call LteRsrp returns the Received Signal Reference Power (RSRP) defined as: [1]

$$\mathrm{RSRP} = \mathrm{ASU} - 140 \qquad [\mathrm{dBm}] \tag{C.2}$$

Unfortunately the SignalStrength API only report updates while the screen is ON, and therefore the CellInfo API is used while the screen is OFF. The CellInfo API was made available in Android SDK version 17 (Android 4.2), but the subclass CellInfoWcdma was not added until SDK version 18 (Android 4.3) [5]. For 3G the CellInfoWcdma is used to extract received power values via the call .getCellSignalStrength().getDbm(). For LTE the subclass CellInfoLte is used with the call .getCellSignalStrength().getAsuLevel() which returns an ASU value defined between 0-97, and where 99 is unknown [1].

During development and measurements we have observed that the behavior of received power values from the CellInfo API varies from phone to phone from different manufacturers in terms of update frequency and availability. This indicates that different manufacturers implement the received power reported to the Radio Interface Layer from the network modem differently, as also observed by [3, 11].

C.3 Methodology

The purpose with this work is to verify that NetMap, using the Android APIs on a commercial smartphone, is able to accurately measure 3G RSCP and LTE RSRP received powers. Our methodology is to compare the NetMap measurements, made in 4 different scenarios, with quality references obtained by use of professional measurement tools consisting of a Rohde & Schwarz radio network scanner, from now on referred as the scanner, and 2 Qualipoc measurement smartphones from SwissQual. The details of the tools and scenarios are given in the following Sec. C.3.1. This allows for the following three comparisons:

1. *Radio network scanner vs. NetMap*
   The scanner has the best measurement resolution and sampling time, but it is a passive device unable to connect to a specific network. The scanner is often favored for drive tests due to its ability to monitor multiple carrier frequencies at once. However, it will not reflect the end-user experience, including handover settings, traffic steering, and cell load conditions, as NetMap will.

2. *Qualipoc vs. NetMap running on the same phone*
   Running NetMap on the measurement phone entails NetMap and the Qualipoc software should report the same received power, because they have the same origin. However, the measurement phone is rooted and the Qualipoc software optimized to provide better resolution and sampling time as compared to the commercial phone.

Table C.1: Key parameters for the phones and measurement tools. RAT specific parameters are given as 3G; LTE.

ID	Model	Software	Android version	Operator & bands	Sampling time [s]	Resolution [dB]
A	Google Nexus 6	NetMap	5.1.1	X: 900,2100; 800,1800,2600	1; 1	2; 1
B	Google Nexus 6	NetMap	5.1.1	Y: 900,2100; 800,2600	1; 1	2; 1
C	Google Nexus 6	NetMap	5.1.1	Z: 2100; 1800,2600	1; 1	2; 1
D	Samsung GS3	Qualipoc 13.0.0.25	4.1.2†	X	0.32; 0.52	1; 0.1
E	Samsung GS5	Qualipoc 15.0.0.53	4.4.4†	X	0.29; 0.53	1; 0.1
F	R&S TSMW radio scanner	Romes 4.82	-	passive X,Y,Z	1.1; 0.11	0.1; 0.01

† SwissQual provided a modified version of Android to run Qualipoc

3. *Qualipoc vs. NetMap running on a different phone*
   Running NetMap on a different phone is expected to result in received power differences, because the two phones will not experience the same fast fading. In addition, the Radio Frequency (RF) front ends and application layers are different. However, since both phones are connected to the same MNO the measurements should be comparable and reflect the accuracy that can be obtained in practice.

C.3.1 Tools and Scenarios

The list of phones and measurement tools as well as their key characteristics are given in Table C.1. To illustrate NetMap's potential to crowdsource coverage, latency, and other network KPIs the application was installed on 3 identical, commercial phones (A,B,C), which were connected to 3 major MNOs (X,Y,Z) in Denmark. The measurement phones (D,E) were connected to operator X, while the scanner (F) passively monitored the received power from all 3 MNOs simultaneously.

As indicated in Table C.1 NetMap's resolution is 10-100 times worse than the professional tools (D-F), and therefore it especially interesting to analyze whether the received power measurements are comparable, because NetMap will then provide a cheap and easily deployable alternative. NetMap's sam-

Table C.2: Scenario details with specific parameters averaged from Qualipoc phones (D,E). RAT specific parameters are given as 3G; LTE.

Scenario	AAU	AAU	Aalborg	Highway
Parameter	indoor	outdoor	city center	E45
Fig. reference	C.2a yellow	C.2a green	C.2c	C.2b
Device speed [km/h]	6 (pedestrian)	6	30	110
Distance [km]	0.44	0.35	2.7	5.7
Observed cells [-]	2; 1	2; 1	14; 12	9 ; 6
Minimum power [dBm]	-100; -110	-87; -96	-96; -114	-106; -117
Maximum power [dBm]	-61; -72	-62; -79	-41; -56	-47; -62

pling time is also lower than Qualipoc's and the scanner in LTE mode. For 3G the scanner, used in high accuracy mode, has a sampling time similar to NetMap because it monitors a large number of bands. NetMap's lower sampling time and resolution may be more of an issue in some scenarios than in others and therefore the 6 devices were deployed in 4 different scenarios; indoor & outdoor at AAU campus, in Aalborg city center, and on the local highway. The details of the scenarios are listed in Table C.2 and they clearly provide different propagation conditions as reflected by the device speed, number of observed cells, and dynamic range of the received signal. The number of observed cells and received powers are based on reports from the Qualipoc phones (D,E). Fig. C.2 illustrate the measurement routes in the 4 scenarios. Note that the yellow line in Fig. C.2a corresponds to the indoor scenario, while the green line is the outdoor scenario.

In order to eliminate effects of the devices moving differently or experiencing different gains due to hand grip effects [2], the devices were mounted in a measurement rack and then moved either by use of a trolley (in the AAU scenarios) or car (city center and highway) as illustrated in Fig. C.3.

C.3.2 Data Processing and Evaluation

After the measurements are completed they are post-processed in Matlab to determine how well NetMap's measurements match those of the Qualipoc phones and the scanner. The processing procedure is as follows:

1. data from the Qualipoc phones and the scanner (devices D-F) are filtered to remove any fast fading effects, because the purpose is to verify whether NetMap captures the overall coverage (mean path loss level) and shadow fading (slow variations of the path loss).

2. data from devices D-F is downsampled (if necessary) to fit the sampling

(a) AAU scenarios

(b) Highway

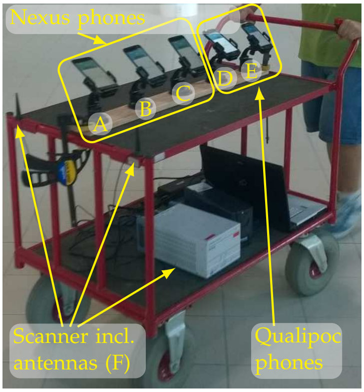
(c) City center

Fig. C.2: The measurement routes in the 4 scenarios.

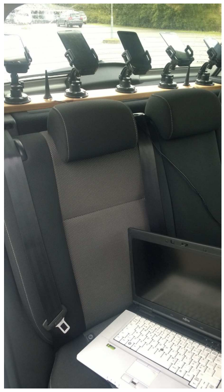
rate of NetMap.

3. comparisons according to the methodologies (1,2,3), described in Sec. C.3, are performed as follows:

    (a) NetMap phones (A-C) are compared with the scanner (F) capturing all 3 MNOs (X,Y,Z). The results are averaged across operators for each of the 4 scenarios. The scanner measures the received power of all cells within its dynamic range while NetMap only measures the received power of the current serving cell. The NetMap measurements are therefore compared with the maximum received power of the scanner, which is determined sample by sample. A hysteresis of 5.5 dB and 3 dB is used for 3G and LTE respectively, to emulate a handover margin between the current serving cell and a stronger neighbor cell.



(a) The trolley



(b) The car

Fig. C.3: The measurement tools and transportation devices.

(b) NetMap measurements of phones D and E are compared with the Qualipoc measurements of the same phones. The results are averaged for the two Qualipoc phone D and E connected to MNO X.

(c) NetMap in phone A is compared with Qualipoc measurements of phones D and E. The 3 phones are connected to the same operator (X), but they may experience small differences in fading, in addition to the different RF front end and antenna gains.

4. the comparisons are based on a parameter search to determine the time- and power-offset between NetMap and reference data. This is necessary because the 6 devices were not started simultaneously and due to the devices' different antenna and RF front end gains.

5. the best fit, depending on the time- and power-offset, is the one resulting in the lowest Root Mean Squared Error (RMSE) and the highest cross-correlation coefficient ρ. Definitions of these metrics are given in the appendix.

Fig. C.4 illustrates the original data after it is time-aligned (thin line) and after it has been filtered (thick line), but not compensated for power-offset. The Fig. also illustrates a potential handover case, where carrier 2 observed by the scanner is stronger than carrier 1. However, during the measurements (not illustrated in Fig. C.4) it was observed that NetMap and Qualipoc phones may be connected to a carrier with lower received power as compared to the maximum value observed by the scanner. The reason is traffic steering and handover policies implemented by the MNO to suit the specific scenario, and differences in antenna and RF front end gains.

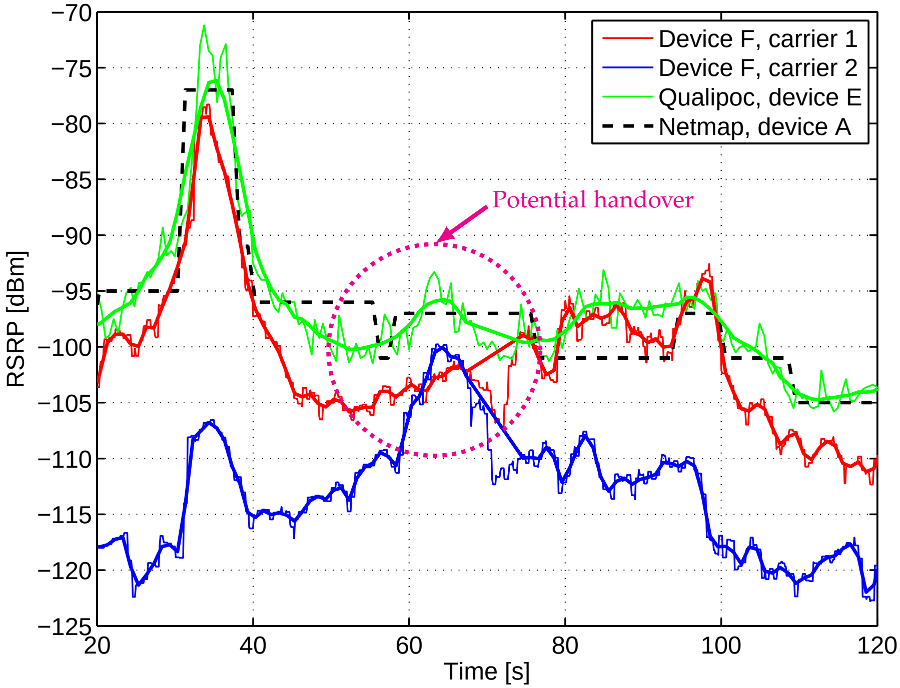


Fig. C.4: Data processing steps. Thin lines are original, time aligned data. Solid lines are filtered and downsampled data. The scenario is indoor LTE.

The power-offset calibration coefficients for NetMap on phone A vs Qualipoc phone D and E are given as an example in Table C.3. In order for the measurements to be valid the power-offset should be constant across the scenarios when compared with a specific device for a specific RAT, because the power-offset only depends on antenna and RF front end gains. The results in Table C.3 reflect this as the standard deviation is around 1 dB for 3G, while it is only about 0.2 dB for LTE i.e. a very constant offset is applied for all scenarios in LTE. In the city center scenario phone A, connected to operator X, performed a handover from LTE to 3G shortly after the measurement was initiated. Therefore the result, marked with *italic*, is unreliable and not included in the calculation of average values. It is not possible to force the phones to LTE, because Voice over IP is not fully implemented in Denmark yet and therefore it would make voice calls to the specific LTE-only phone impossible.

Table C.3: Calibration coefficients for NetMap on phone A vs Qualipoc phones D and E. Values in dB.

Scenario		3G		LTE	
	Phone	D	E	D	E
Indoor		-8.5	-4.8	-0.8	0.7
Outdoor		-8.3	-5.2	-0.5	1.0
City center		-6.2	-5.9	*2.5*	1.6
Highway		-8.5	-4.5	-0.9	0.8
Average		-7.9	-5.1	-0.73	0.83
Standard deviation		1.12	0.61	0.21	0.15

Table C.4: NetMap 3G measurements compared with scanner and Qualipoc.

Comparison	1 (scanner)		2 (same phone)		3 (different phone)	
Scenario	RMSE	ρ	RMSE	ρ	RMSE	ρ
Indoor	4.1 dB	0.52	2.1 dB	0.88	3.7 dB	0.68
Outdoor	4.8 dB	0.57	1.8 dB	0.73	3.3 dB	0.44
City center	6.1 dB	0.72	2.3 dB	0.97	7.0 dB	0.66
Highway	6.8 dB	0.71	4.3 dB	0.92	4.6 dB	0.87
Average	5.4 dB	0.63	2.6 dB	0.88	4.6 dB	0.66

C.4 Results

In this section the RMSE and cross-correlation coefficient ρ results are presented for the 3 comparisons defined in Sec. C.3. The 2 KPIs: RMSE and ρ are defined in the appendix.

The results for the 4 scenarios when using 3G is given in Table C.4. As expected comparison 2 (NetMap and Qualipoc on the same phone) results in the best fit with an average RMSE close to NetMap's resolution of 2 dB (see Table C.1), and a high cross-correlation coefficient of 0.88. The comparisons 1 and 3 with the scanner and Qualipoc running on a different phone yield less accurate results for 3G as the correlation on average is below 0.7 while the RMSE is above 4 dB.

The indoor and outdoor AAU scenarios show the smallest dynamic range of the received power according to Table C.2, and this is reflected in the results in Table C.4 where those scenarios result in the lowest RMSE. However, on average the smaller variations also entail a lower cross-correlation coefficient as compared to the city center and highway scenarios.

The results for LTE are given in Table C.5. As in 3G the comparison 2

Table C.5: NetMap LTE measurements compared with scanner and Qualipoc.

Comparison	1 (scanner)		2 (same phone)		3 (different phone)	
Scenario	RMSE	ρ	RMSE	ρ	RMSE	ρ
Indoor	3.0 dB	0.83	2.1 dB	0.88	2.6 dB	0.87
Outdoor	3.0 dB	0.72	0.8 dB	0.93	2.0 dB	0.67
City center	5.8 dB	0.80	1.9 dB	0.99	4.5 dB	0.85
Highway	7.3 dB	0.74	1.8 dB	0.99	4.5 dB	0.91
Average	4.7 dB	0.77	1.7 dB	0.95	3.4 dB	0.83

provides the best results, but for LTE the comparisons 1 and 3 also provide accurate results with an average cross-correlation coefficient around 0.8 i.e. a good match between NetMap and the professional tools.

Fig. C.5 illustrates the filtered, downsampled, and time- and power-offset results for Qualipoc and NetMap measurements on phone E compared with NetMap measurements on phone A i.e. all connected to the same operator (X). The scenario is indoor LTE. The NetMap measurement on phone E seems to vary slightly more than the NetMap measurement on phone A. Since NetMap was configured to provide one measurement per second for both phones, see Table C.1, the difference is expected to be due to the model and configuration of the chipset and processor.

C.5 Discussion

The results, presented in the previous section, verified that NetMap, running on a commercial smartphone, is able to measure LTE RSRP with sufficient accuracy to track shadow fading and path loss. It provides a cheap alternative to the professional tools even though the resolution and sampling time are significantly lower. In addition, NetMap reports the measurements of a connected phone as opposed to the scanner, which in some cases may overestimate the coverage, because it is not able to capture phenomenons caused by MNO traffic steering. NetMap's 3G measurements are less accurate, partly due to the Android API, but still reliable. The RMSE of 3-5 dB (see Table C.4) is not critical when considering that empirical path loss models and ray-tracing tools compared with received power measurements may result in RMSEs of 4-6 dB [9].

The NetMap measurements were calibrated towards either the scanner or the Qualipoc phones and therefore the power-offset is relative to the antenna and RF front end gain of these devices. This entails the absolute values are not accurate, while the relative measurements are calibrated. This is espe-

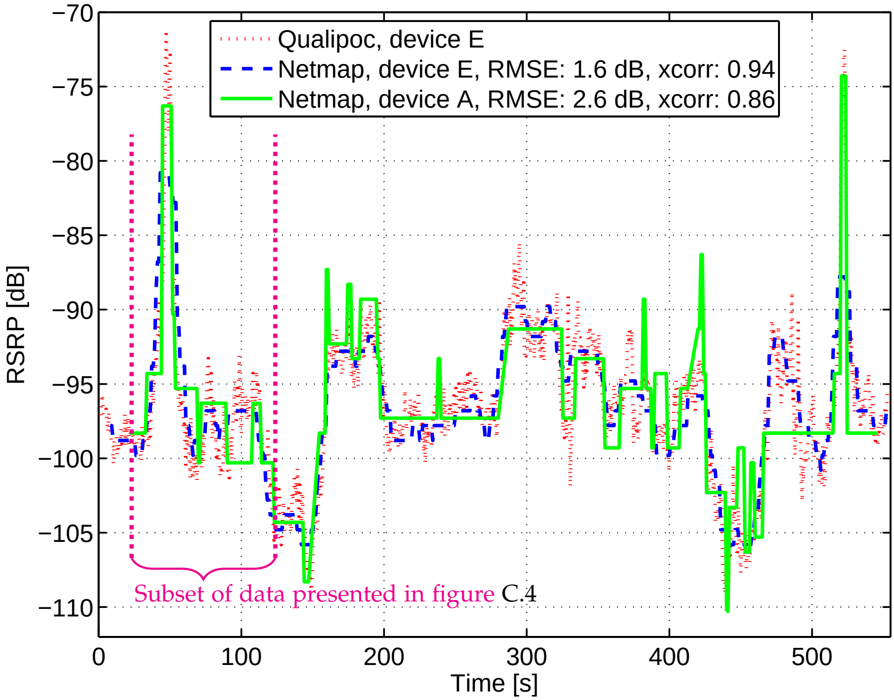
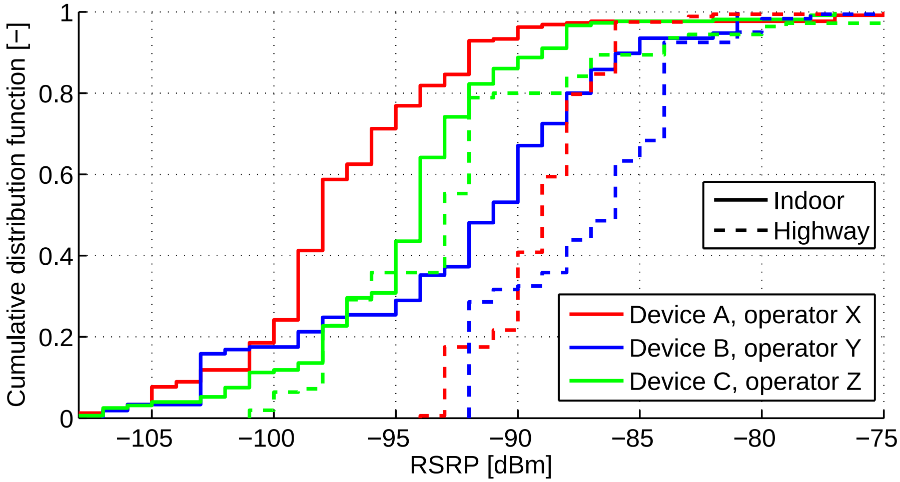


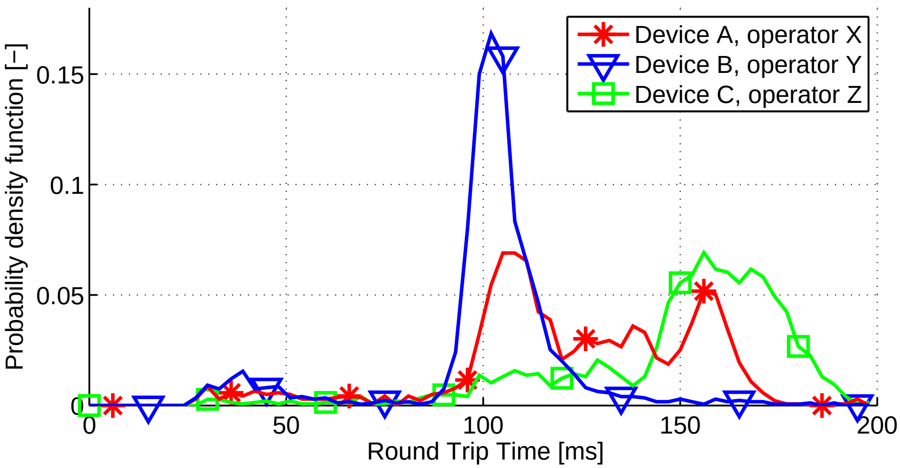
Fig. C.5: NetMap measurements from Qualipoc phone E and commercial phone A compared with the Qualipoc measurement. The scenario is indoor LTE.

cially important for the crowdsourcing results, because as the measurements show the average power-offset in Table C.3 is as high as 8 dB between phones A and D. Thus, there may be significant differences in crowdsourced data from different phones, which must be compensated in the final analysis. This variability was also observed by [3, 11].

Having verified the NetMap received power is an important achievement, because it enables the further analysis of statistics such as latency and throughput, and why those parameters in some cases are worse than expected. In addition, the received power measurements can be used to study and compare the coverage of various MNOs. As an example Fig. C.6a shows the Cumulative Distributive Function of the received power for phones A-C connected to operators X, Y, and Z respectively. The measurements are made for LTE in the indoor AAU and highway scenarios. Operators X and Y seem to benefit from having a sub-GHz carrier in the highway scenario, while operator Y in general provides the best coverage for both scenarios. Fig. C.6b illustrates the combined NetMap LTE RTT measurements for each of the 3 operators averaged across the 4 scenarios. Significant variations can be ob-



(a) NetMap RSRP for LTE indoor and highway scenarios.



(b) NetMap RTT for all LTE scenarios combined.

Fig. C.6: NetMap LTE measurements.

served and if low RTT is of importance to the end-user, operator Y seems like the best choice. Future work includes a correlation analysis of the RSRP and RTT measurements. In addition, the authors of [3] also noted that the use of the signal-to-interference-and-noise ratio metric can be useful, when correlating RTT and throughput measurements with coverage.

C.6 Conclusion

NetMap is an Android application developed for crowdsourcing Mobile Network Operator statistics as observed by the user, e.g. received power, latency, and throughput.

The purpose of this work, being a measurement campaign, was to verify the ability of NetMap to correctly measure received power in 3G and LTE cellular networks. Received power is important when analyzing metrics such as latency and throughput because it affects the modulation and coding scheme that can be applied and the number of retransmissions.

The measurements were performed by connecting commercial smartphones running NetMap to 3 operators in Denmark, while also monitoring the received power using professional measurement phones from SwissQual and a Rohde & Schwarz radio network scanner. The diverse measurement scenarios included indoor & outdoor pedestrian speed traces, and driving on the highway and in the city center of Aalborg.

The results show that NetMap yields accurate LTE measurements with a Root Mean Squared Error of 2-3 dB and cross-correlation coefficient above 0.8, even for high speeds. The 3G measurements result in an error of 3-5 dB and a cross-correlation coefficient of 0.6-0.8, partly due to lower measurement resolution in the Android API. Furthermore, the results show a constant power-offset between NetMap and the professional tools and thus indicate consistent measurements.

Future work includes recording the cell ID concurrently with received power measurements, and presenting the measurement results to the end user, e.g. a coverage map. This will help attract new users, which is vital for crowdsourcing.

Appendix

The Root Mean Squared Error is defined as:

$$\mathrm{RMSE}\left(x,y\right)=\sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(x_{\mathrm{i}}-y_{\mathrm{i}}\right)^{2}}\qquad[\mathrm{dB}]\qquad\text{(C.3)}$$

where x and y are the signal of interest and reference, respectively i.e. a NetMap measurement and a scanner or Qualipoc measurement. The length of the signals is N.

The cross-correlation coefficient ρ is defined as:

$$\begin{aligned}\rho\left(x,y\right)&=\frac{cov\left(x,y\right)}{\sqrt{\sigma_x^2\sigma_y^2}}&&[-]\\&=\frac{E\left[\left(x-\mu_x\right)\left(y-\mu_y\right)\right]}{\sqrt{E\left[\left(x-\mu_x\right)^2\right]E\left[\left(x-\mu_x\right)^2\right]}}&&[-]\qquad\text{(C.4)}\end{aligned}$$

where $cov\left(x,y\right)$ is the covariance of x and y, σ_x^2 is the variance of x, E is the expectation, and μ_{x} is the mean defined as $\mu_{\mathrm{x}}=\frac{1}{N}\sum_{i=1}^{N}x$ for discrete values.

Acknowledgment

This work was supported by Innovation Fund Denmark.

References

[1] 3GPP, "UMTS;LTE AT command set for UE, TS 27.007 v.10.3.0 release 10," April 2011.

[2] E. Buskgaard, A. Tatomirescu, S. Del Barrio, O. Franek, and G. Pedersen, "Effect of antenna bandwidth and placement on the robustness to user interaction," in *Antenna Technology: Small Antennas, Novel EM Structures and Materials, and Applications*, 2014, pp. 258–261.

[3] J. Cainey, B. Gill, S. Johnston, J. Robinson, and S. Westwood, "Modelling download throughput of LTE networks," in *Local Computer Networks Workshops, IEEE 39th Conference on*, 2014, pp. 623–628.

[4] U. Goel, M. P. Wittie, K. C. Claffy, and A. Le, "Survey of End-to-End Mobile Network Measurement Testbeds," *CoRR*, vol. 1411.5003, 2014.

[5] Google Inc., "Developers' reference," Sep. 3 2015. [Online]. Available: http://developer.android.com/reference

[6] N. Haderer, F. Paraiso, C. Ribeiro, P. Merle, R. Rouvoy, and L. Seinturier, "A Cloud-Based Infrastructure for Crowdsourcing Data from Mobile Devices," in *Crowdsourcing*, ser. Progress in IS, W. Li, M. N. Huhns, W.-T. Tsai, and W. Wu, Eds. Springer, 2015, pp. 243–265.

[7] L. M. Mikkelsen, S. R. Thomsen, M. S. Pedersen, and T. K. Madsen, "NetMap - Creating a Map of Application Layer QoS Metrics of Mobile Networks Using Crowd Sourcing," in *Internet of Things, Smart Spaces, and Next Generation Networks and Systems*, ser. Lecture Notes in Computer Science, S. Balandin, S. Andreev, and Y. Koucheryavy, Eds. Springer International Publishing, 2014, vol. 8638, pp. 544–555.

[8] A. Nikravesh, H. Yao, S. Xu, D. Choffnes, and M. Mao, "Mobilyzer: An open platform for controllable mobile network measurements," in *Proceedings of MobiSys*, May 2015.

[9] I. Rodriguez, H. Nguyen, T. Sørensen, J. Elling, M. Gentsch, M. Sørensen, L. Kuru, and P. Mogensen, "A Geometrical-Based Vertical Gain Correction for Signal Strength Prediction of Downtilted Base Station Antennas in Urban Areas," in *Vehicular Technology Conference*, 2012.

[10] S. Rosen, S.-J. Lee, J. Lee, P. Congdon, Z. Mao, and K. Burden, "MCNet: Crowdsourcing wireless performance measurements through the eyes of mobile devices," *Communications Magazine, IEEE*, vol. 52, no. 10, pp. 86–91, October 2014.

[11] S. Sonntag, J. Manner, and L. Schulte, "Netradar - Measuring the wireless world," in *11th International Symposium on Modeling Optimization in Mobile, Ad Hoc Wireless Networks*, May 2013, pp. 29–34.

[12] J. Yoon, S. Sen, J. Hare, and S. Banerjee, "WiScape: A Framework for Measuring the Performance of Wide-Area Wireless Networks," *Mobile Computing, IEEE Transactions on*, vol. 14, no. 8, pp. 1751–1764, 2015.

Paper D

On the Benefits and Challenges of Crowd-Sourced Network Performance Measurements for IoT Scenarios

Lars Møller Mikkelsen, Tatiana Kozlova Madsen, and Hans-Peter Schwefel

The paper has been submitted to the
International Journal on Wireless Personal Communications as a special issue of the selected papers of GWS 2016, 2016.

The layout has been revised.

Abstract

Systems within IoT domains such as ITS, Smart City, Smart Grid and other, often rely on real-time information and communication. These types of systems often include geographically distributed nodes which are connected via cellular or other wireless networks. This means great variability and uncertainty in network connection performance, effectively increasing the expected minimum system response time. Having information about network connection performance means that it is possible to predict the performance of the system in terms of sensor access delay or application response time. We obtain the performance information, in terms of signal strength and transport layer round trip time, using crowd sourcing and consumer devices which causes the measurements to be heterogeneously distributed. From these measurements we want to create a network performance map but in areas with sparse measurements the reliability of the map values will be low. To solve this problem we include neighboring measurements and evaluate the impact of doing so. We show that generally there is a benefit from including neighboring measurements, and that transport layer round trip times are less sensitive to bias when increasing the size of the extended area to include measurements from.

D.1 Introduction

Internet of Things (IoT) is rapidly being developed and starting to being deployed. More and more IoT devices, systems and services are emerging [8]. IoT systems rely on information, and especially information about the world they operate in, such as temperature, air quality, number of users, device state and other. This information can both be used as historical or live information, i.e. from a database with previous values or directly from the sensor as the source of the information. In both cases it is often not enough just to have the information, but also awareness of the quality of the information is needed. One measure of quality could be freshness of the information, or knowledge of what freshness to expect from future updates of the information, i.e. prediction based on historical information.

To be able to predict the freshness of information it is necessary to obtain measurements of end-to-end network delay, which can be obtained cost effectively using crowd sourcing. From the measurements we will create a map of network performance, which represents the geo-dependent cellular network performance by the mean value of different performance metrics; in this paper, we use transport-layer round-trip times and signal strength. In [2] it is shown how a network performance map can be created and used for optimizing TCP based data transfer. To create the map we will divide the geographical area into cells (not to be confused with radio cells) in which we will aggregate the measurements in the mean value. There are cells with only few measurements where the mean estimator is showing a high variance, so it is not 'trustworthy' for further prediction use. In order to reduce the variability, this paper investigates an approach to include neighboring measurements. This increases the number of measurements and therefore reduces variability, but on the other hand may introduce bias, as these are sampled from different locations. This trade-off is analyzed in this paper.

To understand the problem we first have to look at what measurements we include in the map, how we obtain them, and what they will be used for.

In an end-to-end connection in a IoT system between a sensor and an application on a user device, there are at least two wireless links; the sensor connection, and the connection of the user device running the application. In the connection between sensor and application there will also be several wired links, but we assume that the wireless links will by far have the greatest impact on the end-to-end connection performance. The wireless connection of the sensor is typically achieved by using low-power technologies [9], as sensors often are fixed in location and only need to transmit low amounts of data. The wireless connection of the user device will typically be a cellular connection to achieve high mobility and ubiquitous high speed network coverage, but also to support a wide range of usages. We choose to focus on obtaining information about how the cellular wireless link influences the end-to-end connection performance. Subsequent we will denote this as the connection performance.

To get information about the actual connection performance we will apply an active measurement methodology, meaning we will generate measurement traffic and not just utilize already present traffic. Generally there are two approaches that connection performance can be acquired in; dedicated measurements or crowd sourcing. When applying dedicated measurements a very accurate picture is obtained of exactly what is measured, but the measurements can be costly in terms of time and measurement equipment [7]. From the crowd sourcing measurements more unknowns are included in the results which must be handled in post processing [1], while the costs of performing the measurements are low [4]. In this work we use crowd sourcing to measure the connection performance from the end user devices, realized using the NetMap system [6].

The measurements are influenced from factors such as signal disturbances and interferences, network load, device load, different device and antenna characteristics, different networks and network technologies, etc., all of which cause measurement values to vary. This has been explored and documented in works such as [12] that shows that movement highly influences the measured connection performance, while [5] studies the impact of cellular connections content access in general. Furthermore, [10] show that signal strength and higher layer metrics not necessarily are highly correlated, underlining the need for measurement of both type of metrics. The varying measurement distribution is due to the layout of roads and buildings, and how users move and where they spend more or less time. This means that one area can have many measurements, and the immediately neighboring area can have few or no measurements. This is investigate in [11] where a bandwidth map is created from measurements performed only while driving on roads. Furthermore, [3] evaluate the influence of the hidden state of the network, i.e. other factors than location, on network performance is evaluated.

In this work we focus on how to handle the varying measurement density, by evaluating the mean value and the impact on the mean when enhancing sparse measurement cells with measurements from neighboring cells.

The rest of this paper is organized as follows; in Section D.2 the measurement method and the measurements are introduced. Section D.3 describes the approach we apply for evaluation of the measurements and presents the evaluation results. Section D.4 concludes on the results and gives and outlook to future work.

D.2 Crowd-Sourced Network Performance Measurements

We will base our evaluation on two measurement sets, one obtained in an urban area, and one in a rural area. In this section we first describe the measurements and how they are obtained, followed by a description of the processing approach and evaluation of the results.

As mentioned in the introduction we have decided to focus on the performance of the cellular connection, because typically this is the link with the highest influence on the end to end connection performance as seen from the user perspective. The measurements will be performed by consumer smartphones using the crowd sourcing measurement system NetMap.

D.2.1 Measurement collection software and metrics

The measurements were collected using the NetMap system [6] on Android devices, and are all performed using a 3G connection. NetMap is a system developed for performing crowd based network performance measurements. Users install an app that periodically performs measurements of various QoS metrics on the cellular data connection, and automatically submits the results to the back end, along with a wide range of additional context information about the device at the time of measurement.

For our purpose the QoS metrics we measure are packet round-trip time (RTT) and signal strength. The measurement system distinguishes between the RTT of TCP packets and of UDP packets. Both times are measured while actively exchanging data between the mobile device and a measurement server. The signal strength values are the result of a passive measurement that does not include sending any data, but are measured while the connection is active measuring RTT. Consequently each individual sample contains either a TCP RTT or a UDP RTT value along with a signal strength value, a timestamp, and longitude and latitude of the location.

RTT is recorded as the time it takes to send a data packet from the client to a server and to send a data packet from the server to the client, where the server replies as fast a possible. This is done with 20 request/reply sequences both for TCP and UDP. For TCP, the connection is initialized before the measurement starts. For both TCP and UDP the client waits for the reply to the previous request to arrive before sending the next request, and the size of the data is 20 bytes.

As soon as the sampling process has been started in the NetMap app, measurements are done periodically until the process is manually stopped. The schedule of measurements is configured such that in each round first TCP RTT is measured together with signal strength, and then, after a delay of 2 seconds, measurements of UDP RTT and signal strength are performed. The process then remains idle for a uniform random time of 0-10 seconds before the next round starts.

D.2.2 Measurement setting

We collected measurements in two different settings: rural and urban. In the urban setting we collected measurements in an area confined to a few streets with a mix of residential buildings and shops, while walking in a normal pace. The measurements were performed using 2 similar devices (LG G4c (LG-H525N) and LG G3 (LG-D855)), both connected to the same network, using 3G UMTS 2100 MHz as connection technology.

In the rural setting we collected measurements on a 13km stretch of road going through a rural area. The measurements were collected both while driving and walking. The measurements were performed using 2 identical devices (Motorola Nexus 6), both connected to the same network, using 3G UMTS 900 and 2100 MHz as connection technology. In both scenarios the measurements were collected during several days, but only between 8 in the morning and 8 in the evening.

D.2.3 Measurements

Table D.1 lists the amount of measurements collected in the two settings.

In Figures D.1 and D.2 it can be seen where the measurements were collected in the two scenarios.

In Figures D.3 and D.4 the measurement value distributions of measurements from the two settings can be seen.

Measurement type	Rural	Urban
TCP RTT	1586	475
UDP RTT	1580	477
Signal Strength	3166	952

Table D.1: Number of measurements per type and per setting.

Fig. D.1: Locations of urban measurements map on Google Maps.

D.3 Evaluation of Measurements

In this section we describe how we create a cellular network performance map and how we employ a simple interpolation approach to handle sparse measurement cells. Furthermore, we describe our approach to evaluate the impact on the map values when interpolating measurements to sparse measurement cells.

D.3.1 Measurement Processing Approach

Here we describe how we generate a network performance map, and from that highlight the problem we investigate. We will refer to the area where we have measurements within as the full geographical area. The full geographical area is divided into square non-overlapping cells (not to be confused with radio cells), and from the measurements within each cell we calculate the sample mean and a confidence interval of the sample mean. The chosen size of the cells, and thereby the resolution of the map, will depend on

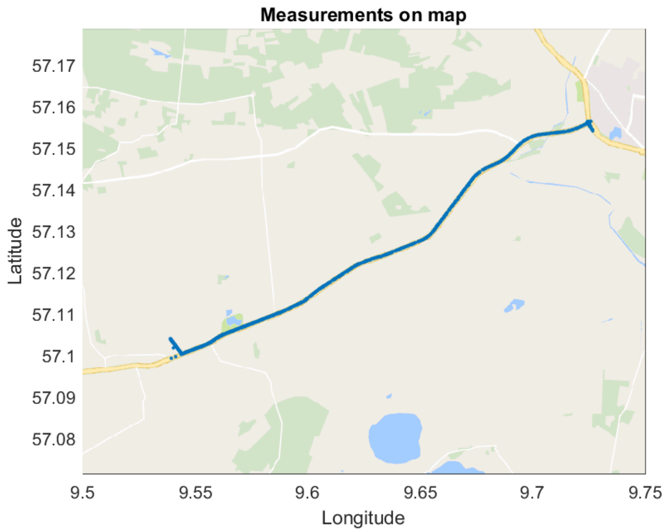


Fig. D.2: Locations of rural measurements on Google Maps.

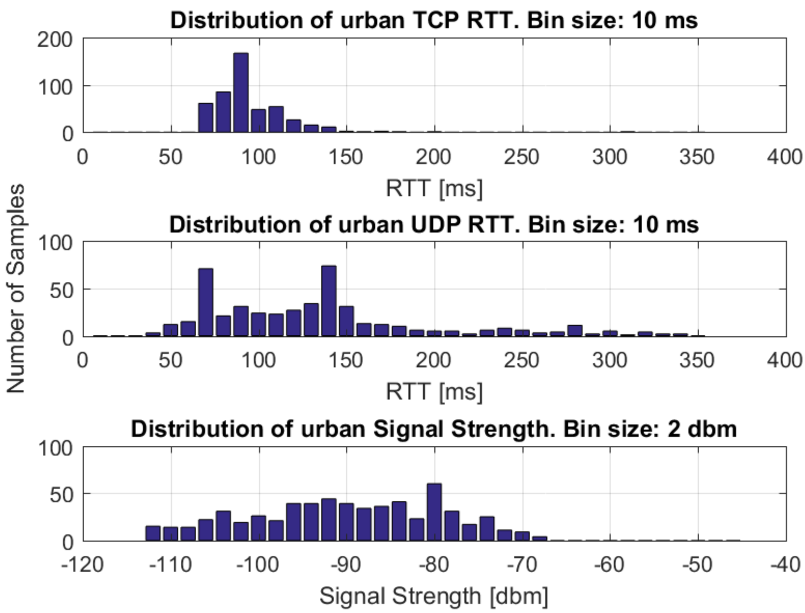


Fig. D.3: Measurement distributions of full urban dataset.

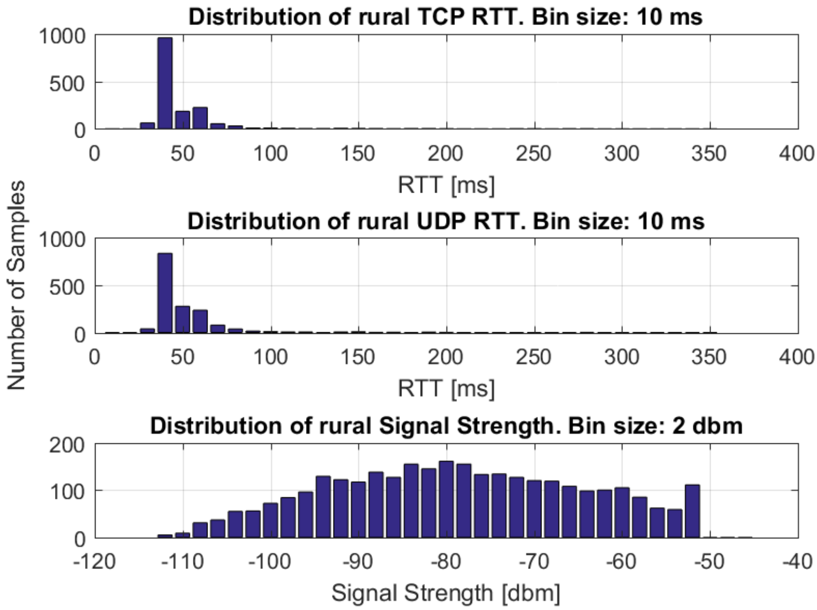


Fig. D.4: Measurement distributions of full rural dataset.

requirements of the use case of the map, which we will not specify further, but we will evaluate cell sizes between 20m and 65m. Depending on the cell size and measurement density, we will have cells with statistically sufficient measurements and cells with less. We select 30 measurements as a minimum to ensure a good statistical basis for the sample mean. In cells with fewer than 30 measurements we interpolate neighboring measurements by including them in the sample mean calculation. Adding more measurements will reduce the variance of the mean estimator, but on the other hand may add bias as the additional measurements may be subject to different environment influences than the measurements in the initial cell. We will analyze this impact in the following.

D.3.2 Impact Evaluation Approach

Here we describe how we evaluate the impact of including neighboring measurements when having a sparse measurement cell. The goal is to evaluate the impact as a function of the distance to the included measurements. We assume a given cell size, which however may vary depending on the performance map and the application using it. Practically we will evaluate the impact at different cell sizes.

We start with selecting a cell as the initial cell containing a minimum of 30 measurements. We denote the measurements within the initial cell as m_S. We calculate the sample mean of measurements in this cell and denote this as the ground truth (GT). Furthermore, we calculate the 95% confidence interval of the GT (CI_{GT}) based on the measurements. CI_{GT} will be the basis of the further evaluation. In our evaluation we will for simplicity define cells as circles, where the radius of the circle is D. We will evaluate cells with diameter between 20m and 65m.

To simulate a cell with sparse measurements we sample m_S to get n=20 subsets, S_n (see Figure D.5). Each of these subsets will contain 5 measurements randomly selected from m_S. We calculate the mean of each of these subsets, which we will call μ_n.

We now start to include neighbor measurements in the subsets from outside the initial cell. We do this by extending the initial cell by defining a radius R, where $R \geq D$ and include the measurements placed outside the initial cell with radius D, and inside the extended cell with radius R. We call this set m_R. We include measurements by combining the full m_R with each

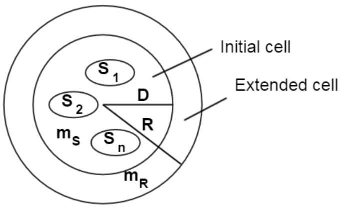


Fig. D.5: Representation of sampling approach.

of S_n subsets, which gives us new subsets $\hat{S}_n$. For each of the new subsets we calculate the mean $\hat{\mu}_n$.

Now we evaluate for each $\hat{\mu}_n$ if it is similar to GT, by checking if it is within CI_{GT}. If inside we give it the indicator value of 1 and 0 if outside. By averaging the indicator values over all subsets we get the average similarity as a value between 0 and 1. We repeat this for several values of R giving the similarity between the enhanced subset means $\hat{\mu}_n$ and GT.

Following are listed the relevant parameters for the evaluation:

- D: Initial cell radius. In the range of 20m to 65m
- R: Extended cell radius to include further measurements within. In the range of D to 180m (300m for rural)
- Measurement types evaluated: TCP RTT, UDP RTT, and Signal Strength

- n: n=20 subset samples from the initial cell
- Subset sample size: 5 randomly selected measurements as a subset of m_S

Example of initial cell with sparse measurements: In Figure D.6 we see an example of the processing of UDP RTT measurements from urban setting in an initial cell for D=20m, where R is in the range of 20m to 180m. In

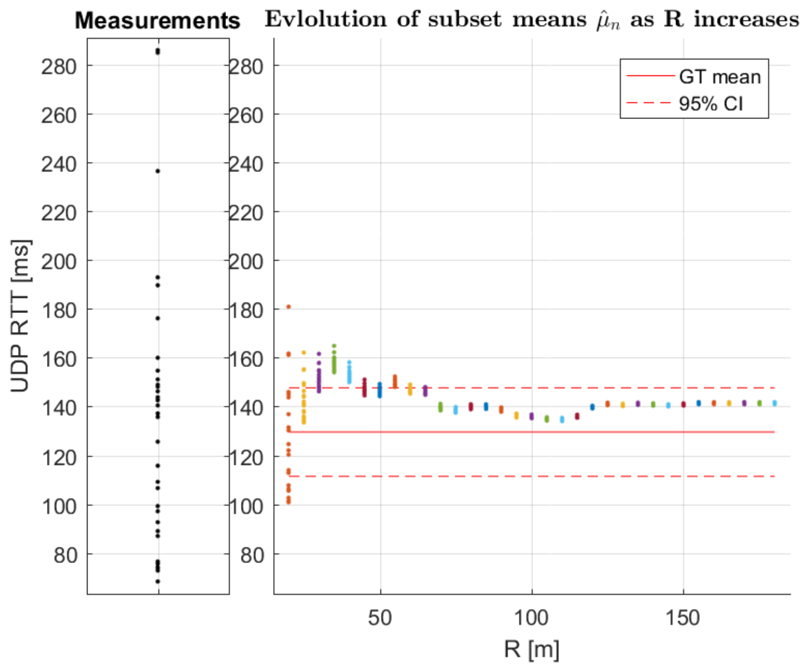


Fig. D.6: Example of an initial cell in urban setting with D=20m, and the processing of UDP RTT measurements.

the left plot we see the individual measurement values from the initial cell (m_S). In the right plot we see the evolution of the subset means ($\hat{\mu}_n$) from R=20m up to R=180m. At R=20m no measurements outside the initial cell are included in the subsets while as R increases more and more measurements are included in the subsets from the extended cell. As R increases and we include more and more measurements (m_R) in the subsets, and in effect the means become less and less spread out. In this example we have relatively high spread in the measurements in the initial cell, which leads to spread in subset means for low R. We can see that for low R some of the means are inside CI_{GT}, and some are outside. In this initial cell from *R=75m onwards for the investigated range until R=180m the subset means happen to all stay within CI_{GT}, but for another initial cell it could happen that they are outside CI_{GT}.

D.3.3 Impact Evaluation

Now we will analyze the impact of including neighboring measurements in sparsely populated measurement cells. We do this based on the output of our evaluation approach, described in previous section. Figures D.7 through D.12 show the evaluation output. We have evaluated results for all integer values of D between D=20m and D=65m, but to make the plots easier to read we only show results from a subset of D values. The conclusions do however still hold as the graphs evolve gradually from low to high D values. Note the graphs shows the similarity indicator averaged over several initial cell.

Rural and Urban Signal Strength Evaluation

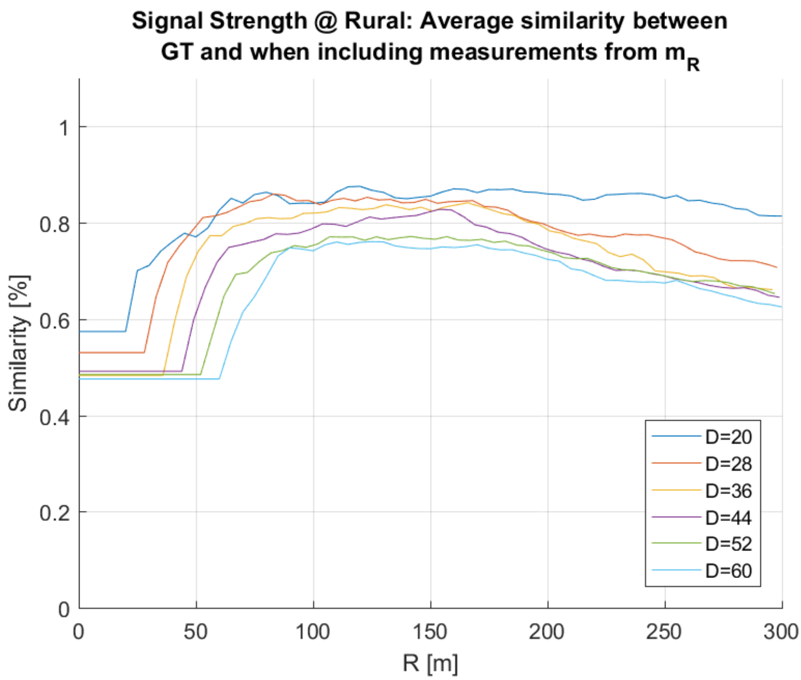


Fig. D.7: Subset means similarity to GT of Signal Strength measurements in rural setting.

In Figures D.7 and D.8 we see the similarity between GT mean and the means of sparse measurements sets, when including measurements from m_R

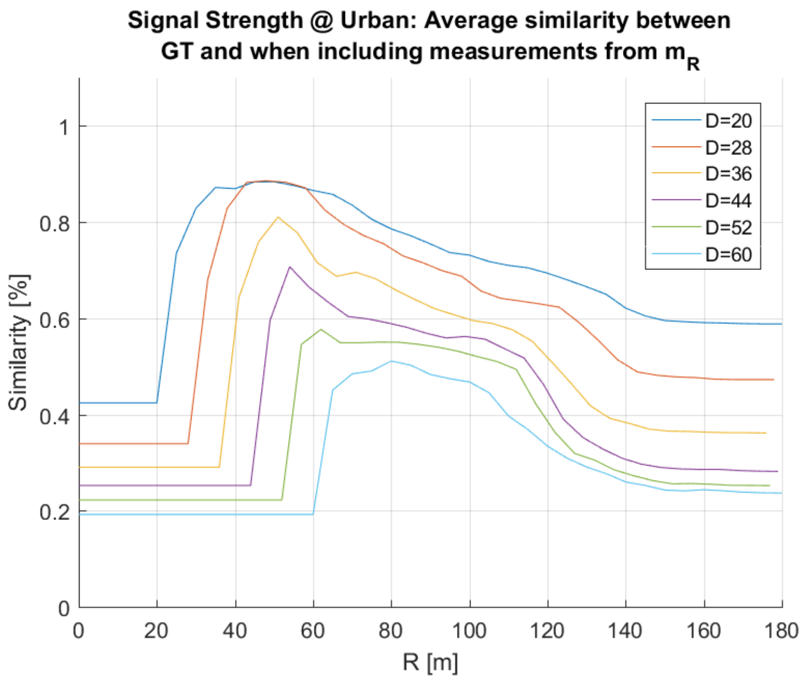


Fig. D.8: Subset means similarity to GT of Signal Strength measurements in urban setting.

for Signal Strength measured in rural and urban settings. For both measurements in the rural and urban setting we see the similarity graphs increase when we include neighboring measurements. For rural setting the graphs rise to a maximum value between 75% and 85%, and around R=165m they start decreasing again. For urban setting the graphs rise quickly to between 50% and 80%, and starts to decrease immediately after the initial increase. This drop continues to around R=140m after which the graphs evens out. Besides the decrease behavior after the initial increase another difference between measurements from rural and urban setting is the max similarity values that the graphs rise to initially. For urban setting the max level seem to be dependent on the D value and starting similarity.

Signal Strength Measurements Impact Considerations

From the Signal Strength measurements there is a clear indication of the initial benefit when including neighboring measurements in sparse measurement sets. But as we increase the size of the cell where we include measurements from the similarity decreases, with the decrease being faster for urban setting than for rural. This indicates that the wireless signals change much faster in urban setting than in rural setting. This makes sense as in urban setting there are more obstacles to signal propagation, i.e. turning around a corner of a building will give you a significantly different signal path. So the maximum distance from which we should include neighboring measurements depends on the area type.

Urban TCP and UDP RTT Evaluation

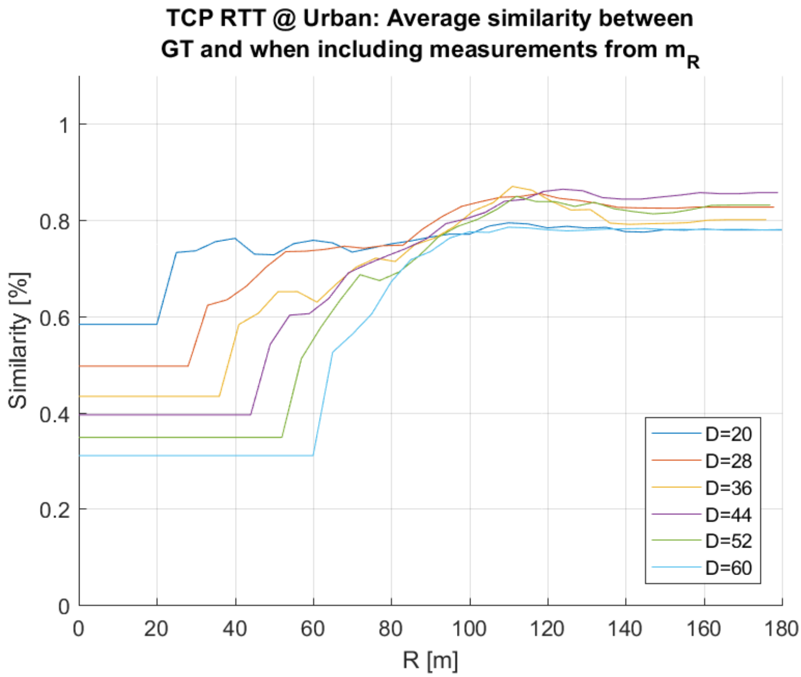


Fig. D.9: Subset means similarity to GT of TCP RTT measurements in urban setting.

In Figures D.9 and D.10 we see the similarity between GT mean and the means of sparse measurements sets when including neighboring measurements, for both TCP and UDP RTT measured in the urban setting. We see that for both TCP and UDP the similarity increases as soon as we start including neighboring measurements from m_R. For TCP RTT the maximum similarity values are between 75% and 85% while for UDP RTT the maximum values are between 60% and 75%. Furthermore, the similarity graph for the smallest value of D for UDP RTT experience a drop after the initial increase before rising to the maximum value.

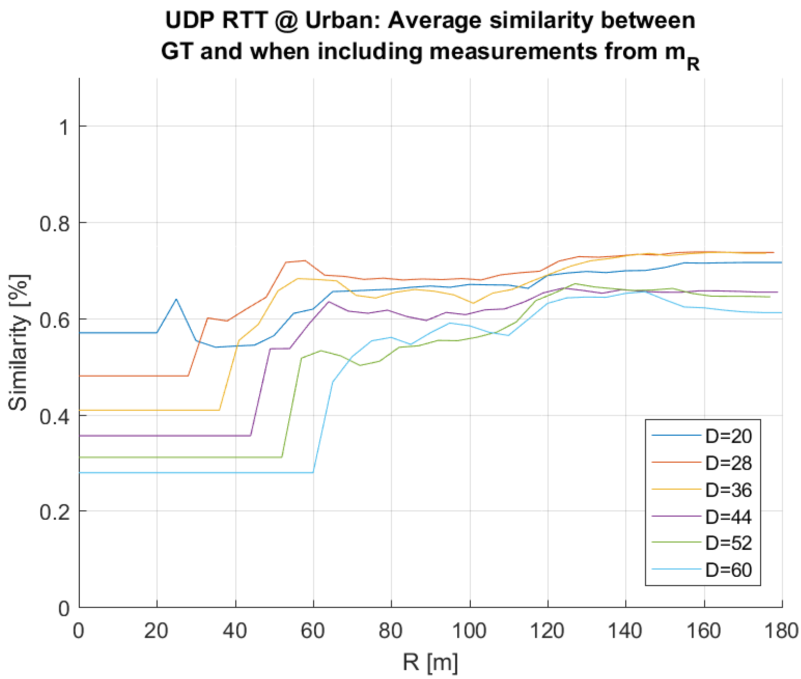


Fig. D.10: Subset means similarity to GT of UDP RTT measurements in urban setting.

Rural TCP and UDP RTT Evaluation

In Figures D.11 and D.12 we see the similarity between GT mean and the means of sparse measurements sets when including neighboring measurements, for both TCP and UDP RTT measured in the rural setting. Again here for both TCP and UDP RTT the similarity graphs increase as soon as measurements are included from m_R. For TCP RTT the maximum similarity for small values of D is at 100%, while for bigger values of D the maximum similarity values are around 80%. For UDP RTT the maximum similarity for small values of D is also at 100%, while for bigger values of D the maximum similarity is between 90% and 95%. Again here for D=20m, as it was the case for UDP RTT in urban setting, the similarity shows an initial drop, before rising to 100%.

TCP and UDP RTT Measurements Impact Considerations

For UDP and TCP RTT there is also a clear benefit of including neighboring measurements in sparse measurement sets. But what is different here from the Signal Strength measurements is that the benefit does not seem to disap-

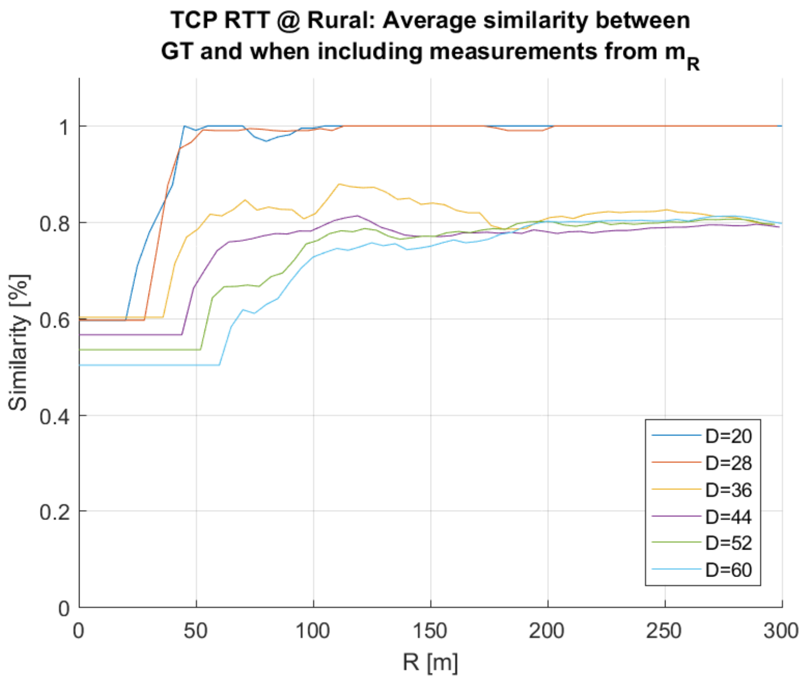


Fig. D.11: Subset means similarity to GT of TCP RTT measurements in rural setting.

pear when including measurements from further and further away, or at least not within the distance in the available data. This means that if we obtain statistically sufficient measurements by including measurements within the first 20-40m outside the initial cell, then there is no additional benefit in similarity by looking further away, or at least not within 180m and 300m for urban and rural setting respectively. This is because for large R the measurements will always introduce bias as it will pull the mean in the limit to the mean value over the whole space.

In both rural and urban setting for TCP RTT the maximum similarity is around 80%, with an exception of small initial cell size for rural setting, which evens out at 100%. For UDP RTT in rural setting the maximum similarity value is higher than for urban setting. These observations for TCP and UDP RTT can be explained by looking at the distribution plots of the urban and rural setting measurements in Figures D.3 and D.4. For rural setting the TCP and UDP RTT both have more narrow distributions than for urban setting measurements, why the maximum similarities are higher for rural than for urban setting. Furthermore, the TCP RTT distributions for urban and rural settings are more similar than the UDP RTT distributions, why the similarity

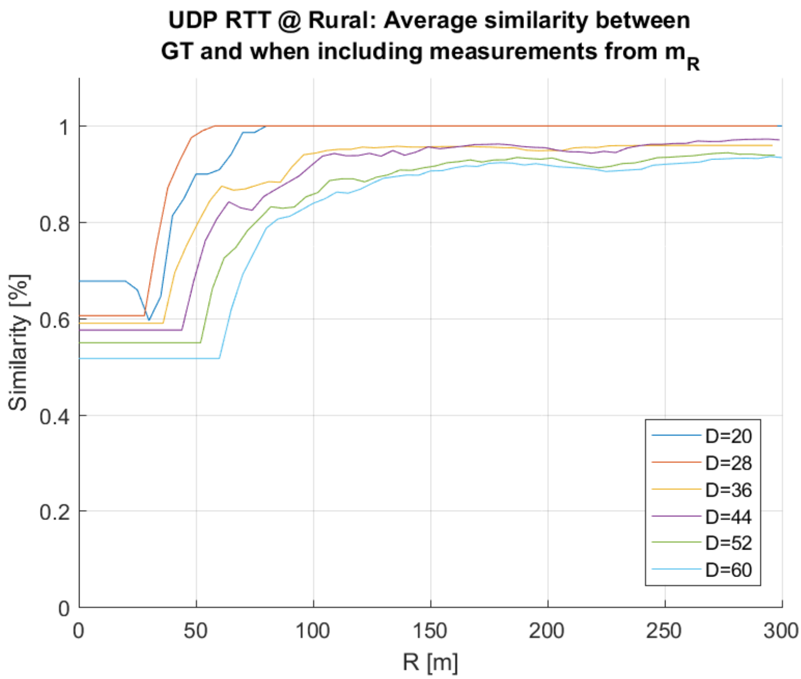


Fig. D.12: Subset means similarity to GT of UDP RTT measurements in rural setting.

graphs look more alike for TCP RTT than for UDP RTT.

D.3.4 Recommended Size of R

Based on the evaluation in the previous paragraphs we can now make some recommendations of how far away from the sparsely measurement populated cell, or initial cell, to include measurements from.

For Signal Strength measurements in urban setting we can recommend to only go 20m-40m outside the cell as after that the benefit is reduced. In rural setting the distance is greater going up to around 100m outside the cell, as further away we see a small decrease in the benefit.

For TCP and UDP RTT measurements the recommendation is not as strict because we do not see a decrease in similarity values within the range of distance values that we investigated in the experiments. We would advice to not increase the cell further when statistically sufficient measurements have been obtained due to the introduction of bias.

D.4 Conclusion

In the previous sections we evaluated the impact on the mean estimate from cells with sparse measurements when including neighboring measurements, by comparing to GT mean of the cell. We did this by evaluating the similarity between means of subsets with GT mean. Increase in the similarity indicates that subset means are improved on average, i.e. more of the subset means are inside CI_{GT}. Generally when including neighboring measurements the similarity increases. This seem true for both Signal Strength and TCP and UDP RTT, and both in rural and urban settings. But limits to the distance to the included measurements vary depending on measurement metric and setting.

From this we can conclude that we can enrich the sparse measurement sets without compromising the accuracy of the mean estimate. For RTT, which is a transport layer performance metric, there seems not to be any significant impact of including neighboring measurements up to 180m or 300m for urban and rural setting respectively. For Signal Strength however, there seems to be a limit to how far away we can include measurements from, where the measurement setting is the defining factor.

Outlook In this paper we focused on the distance outside the initial cell, while not considering the size of the initial cell. This is however also an interesting topic to explore, as different use cases will have different requirements to map resolution. So we would like to investigate if it makes sense to make small cells in the map, or if there is a minimum limit.

In this paper we looked at the cell by cell impact of including neighboring measurements, but looking further ahead we would like to investigate the impact on a network performance map. This could be in terms of impact on detail level and coverage level.

Furthermore, we would also like to look at the actual use case of the network performance map, i.e. using it to attach quality metrics to transfered information in IoT.

Acknowledgment The work is supported by BIG IoT (Bridging the Interoperability Gap of the Internet of Things) project funded by the European Union Horizon 2020 research and innovation programme under grant agreement No 688038.

References

[1] A. Faggiani, E. Gregori, L. Lenzini, V. Luconi, and A. Vecchio, "Smartphone-based crowdsourcing for network monitoring: Opportunities, challenges, and a

case study," *IEEE Communications Magazine*, vol. 52, no. 1, pp. 106–113, January 2014.

[2] K. Højgaard-Hansen, T. Madsen, and H.-P. Schwefel, *Reducing Communication Overhead by Scheduling TCP Transfers on Mobile Devices using Wireless Network Performance Maps*. IEEE Press, 4 2012, pp. 1–8.

[3] ——, *Addressing the Influence of Hidden State on Wireless Network Optimizations using Performance Maps*. Springer, 2015, pp. 177–189.

[4] J. a. Howe, "The rise of crowdsourcing," vol. 14, pp. 1–4, 2006. [Online]. Available: https://www.wired.com/2006/06/crowds/

[5] F. Kaup, F. Michelinakis, N. Bui, J. Widmer, K. Wac, and D. Hausheer, "Assessing the implications of cellular network performance on mobile content access," *IEEE Transactions on Network and Service Management*, vol. 13, no. 2, pp. 168–180, June 2016.

[6] L. Mikkelsen, S. Thomsen, M. Pedersen, and T. Madsen, "Netmap - creating a map of application layer qos metrics of mobile networks using crowd sourcing," in *Internet of Things, Smart Spaces, and Next Generation Networks and Systems*, ser. Lecture Notes in Computer Science, S. Balandin, S. Andreev, and Y. Koucheryavy, Eds. Springer International Publishing, 2014, vol. 8638, pp. 544–555. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-10353-2_50

[7] A. Misra, A. Gooze, K. Watkins, M. Asad, and C. Le Dantec, "Crowdsourcing and its application to transportation data collection and management," *Transportation Research Record: Journal of the Transportation Research Board*, no. 2414, pp. 1–8, 2014.

[8] R. Porkodi and V. Bhuvaneswari, "The internet of things (iot) applications and communication enabling technology standards: An overview," in *2014 International Conference on Intelligent Computing Applications*, March 2014, pp. 324–329.

[9] S. M. Rakshit, M. Hempel, S. Banerjee, and H. Sharif, "Towards a modular iot network model: Low power radio domain use-case in linear-topology wireless sensor network," in *2016 International Conference on Selected Topics in Mobile Wireless Networking (MoWNeT)*, April 2016, pp. 1–7.

[10] S. Sonntag, L. Schulte, and J. Manner, "Mobile network measurements - it's not all about signal strength," in *2013 IEEE Wireless Communications and Networking Conference (WCNC)*, April 2013, pp. 4624–4629.

[11] J. Yao, S. S. Kanhere, and M. Hassan, "Improving qos in high-speed mobility using bandwidth maps," *IEEE Transactions on Mobile Computing*, vol. 11, no. 4, pp. 603–617, April 2012.

[12] J. a. Yao, "Proc. acm workshop wireless netw. testbeds exp. eval. charact." pp. 11–18, 2008. [Online]. Available: http://sfx.aub.aau.dk/sfxaub?url%5fver=Z39.88-2004&rfr%5fid=info:sid/IEEE.org:XPLORE&url%5fctx%5ffmt=info:ofi/fmt:kev:mtx:ctx&rft%5fval%5ffmt=info:ofi/fmt:kev:mtx:journal&rft.jtitle=Proc.ACMWorkshopWirelessNetw.TestbedsExp.Eval.Charact.&rft.date=2008&rft.spage=11&rft.epage=18&rft.au=J.Yao&rft.aulast=J.Yao

Paper E

Accurate and Quality-Aware Bus Occupancy Estimation Utilizing Probabilistic Models for WLAN Probing

Lars Mikkelsen, Hans-Peter Schwefel, and Tatiana Madsen

The paper has been submitted to the
Journal IEEE Transactions on Vehicular Technology, 2017.

The layout has been revised.

Abstract

In this work the goal is to obtain an improved estimator and quality indicator for the number of people on the bus. This is done by estimating the number of WiFi enabled devices, which is the basis for extrapolating the number of people. The estimated number of devices is subject to both false positives and false negatives. The false positives are caused by detecting devices on the roadside outside the bus and not being able to distinguish them from inside bus devices. The amount of false negatives depends on probe emission frequency, message losses due to collisions, MAC address randomization, WiFi channel selection and the time a person stays on a bus. The model proposed in the paper includes the influences of these factors assuming FP and FN being binomially distributed. Distribution parameters are found from the measurements. Furthermore, a quality indicator for the developed estimator is designed. The quality indicator can be useful in practice when selecting and/or fusing data from multiple sources.

E.1 Introduction

Optimization of public transport services can be achieved having real time information about number of passengers. Knowing the amount of people that are transported can provide the transport companies with precious load information to allow them to rearrange resources. For users of public transport, information about the passenger load is also of interest in order to avoid the highly loaded trains and buses. This type of information can be obtained in various ways, e.g. using visual inspection, checkpoint counting, or by inference from communication systems. Different approaches have different advantages and disadvantages in terms of accuracy, deployment and runtime costs, and legal restrictions. Visual inspection can be done by counting the number of people based on camera feeds, which could problematic due to legal restrictions on surveillance in public spaces. Checkpoint counting can be done by setting up sensors at bus or train doors, or at entrances to stations. This might be costly to deploy the wast amount of sensors that is needed. It is possible to estimate the number of devices present based on passively sniffing communication from e.g. WiFi or Bluetooth enabled devices. From the estimated number of devices it is then possible to estimate the number of people. The pitfall is however the inaccuracy of the estimate as it depends on the penetration of the communication system used and the rate of people carrying a device with the technology activated.

As mentioned the inference from communication systems has some disadvantages in terms of inaccuracies, however it still has a clear advantage over the other approaches, namely implementation cost when considering coverage. By exploiting a widely adopted wireless communication technology it is possible to provide people density estimation covering large areas at low cost.

In this work we describe a method for estimating number of people based on an estimate of number of devices. We obtain the estimated number of devices based on WiFi probes collected via a WiFi probe collection system with sensors deployed on buses. The sensors are deployed on buses to support the main use case, which is to estimate bus occupancy. The motivation for obtaining this information is to use it in various use cases. Bus company operators can use the information about bus occupancy to organize and plan resource usage. Public transport users can use the information, e.g. via smart travel assistant apps, to get advice of which bus line to take to avoid the crowd. A by-product of the estimate is information about number of people in areas where the bus pass through. This can be used by the city administration for organizing service personal e.g. in connection with city wide events.

Smart devices supporting Bluetooth and WLAN communication are gaining higher market penetration. This clearly gives opportunities for exploiting the technologies in terms of signaling messages in determining people density. For cellular technologies similar approaches have been applied in works such as [14], [7], [6]. These approaches however require access to internal information from the cellular networks, which is unfeasible in our scenario.

WiFi enabled devices use WiFi probes to search for access points, similarly for Bluetooth when performing discovery as shown in [10]. This can be exploited to estimate the number of people in an area. Approaches such as these allow for counting of number of people carrying WiFi or Bluetooth enabled devices, given that the communication technology is activated, meaning that devices are not counted if the technology is disabled or not supported. This means that the distribution of devices that support these technologies is of interest. Cisco provides us with some information on this matter [9], predicting that 53% of IP traffic in 2019 will be generated from WiFi enabled devices. Furthermore, the number of WiFi hotspots will grow from 64.2 million in 2015 to 435.2 million in 2020. These numbers supports the applicability of estimating number of people based on information extracted from the WiFi communication system.

In the discovery procedure for Bluetooth devices send out requests periodically to discover nearby devices, and given that a nearby device is not in a hidden state it will reply the discovery request. In other literature several implementations of systems for counting Bluetooth enabled devices have been presented. A scanner device is developed and deployed in [15] to collect discovery requests and specifically the MAC address of the emitting device. The collected data is used for creation of origin/destination matrices to improve bus utilization. Similarly in [21] the approach is considered for counting spectators of a European soccer championship.

Today WiFi is even more widespread in usage than Bluetooth, making it more interesting in detecting device presence. In [13] WiFi probes are utilized to estimate crowd density and flow at a major German airport. Sensors are placed before and after the security checkpoint where they scan for WiFi probes. In this scenario they use the number of boarding passes scanned by the personal as the ground truth, allowing the fine tuning of the number of people estimation. [12] presents the use case of counting people on buses. Here it is argued that if passengers are provided with the information about public transport load, they will optimize their usage of public transport to achieve higher comport. In [12] they focus on obtaining high accuracy estimates in as close to real time as possible. However, the accuracy of the proposed approach is not evaluated, and the authors note that the ground truth, i.e. number of active devices, that is needed for accuracy evaluations, has been difficult to obtain. In both [13] and [12] they obtain estimated for number of people in the respective scenarios. But they do not focus a lot on the connection between number of devices and number of people, rather they assume that the obtained estimate is equal to a fixed percentage of the true number of people.

The use of sensor information in distributed systems is subject to two types of inaccuracies degrading information quality: Measurement errors at the sensor may propagate through the whole computation chain, see, e.g., [5] for work characterizing such errors and their impact. For distributed real-time systems, a second cause of inaccuracies requires attention: while the sensor data is being transmitted and processed, the actual physical value changes so that it deviates from the value used in the processing [4]. The benefits of the latter quality metrics has been demonstrated for use cases of adaptive context information access [20] and for location-based relaying [18].

In this paper, we focus on the first type of error, where however the derived information is not directly obtained from sensors, but results from different processing of information from WLAN probing sensors. We will consider a similar use case scenario as in [12], i.e. estimating the number of people on the bus, but we will focus more on evaluation of the estimator. We will first obtain a base estimator of the number of devices on the bus, which is similar to the estimate obtained in [12]. We will then improve on the base estimator by probabilistically modeling the estimator in terms of false positives and false negatives. We will stochastically describe the link between number of devices and number of people, to be able to obtain a people density estimate. We will evaluate the estimate using manually counted ground truth number of people on the bus. Finally we will design a quality indicator of the estimate, that we will attach to the estimate for use by services.

This paper builds on the conference publication [16]. Compared to this conference publication this paper includes the improved estimator, modeling of false positives and false negatives, modeling of multiple devices per person, and design of the quality indicator.

The rest of the paper is outlined as follows. Section E.2 describes the WiFi probe collection system. Section E.3 covers the baseline algorithm and the estimation results of this. Section E.4 describes the overall method applied in modeling the improved estimator. Section E.5 describes how the model parameters are obtained via experimental measurement setups. Section E.6 describe how the link is made between estimated number of devices and number of people. Section E.7 presents the results of evaluating the improved estimator using ground truth observations. Section E.8 describes the design of the quality indicator. Section E.9 the results are summarized and an outlook is given.

E.2 Design of WiFi Probe Collection System

In this section we will describe the design of the WiFi probe collection system in terms of the various components and their main functionalities.

In Figure E.1 the architecture of the system is presented. From this we see that the system consists of several modules, but in this work we will only address the modules marked in green; Processing Service, Collector Server, and Sensor Node. We place the Sensor Node on a bus where it will collect WiFi probes from active WiFi enabled devices on the bus and outside the bus. Each WiFi probe contains a MAC address of the device that emitted the probe. This is anonymized on the Sensor Node before submitting the collected probes to the Collector Server. The Collector Server stores the probes and makes them available to the Processing Service, which fetches the probes and process them using a filtering algorithm.

In this system design the collector server will act as a gateway where multiple sensors nodes can submit collected probes to. Because of this we can scale up the number of Sensor Nodes easily. Similarly, if we want to process the collected probes in a different way, or use them in another context, we simply add another service on top.

This setup is particularly beneficial in the settings of IoT where sensors collect information that potentially is used in several different application domains. Such sensor and platform setups is common in the IoT domain as exemplified in [19]. This does however make the collector server a single point of failure, and in the future it should be replicated to ensure availability.

It can be seen from the figure that the Collector Server pushes the collected probes to an IoT platform which then makes the probes available to services. The services make the processed data available to apps, e.g. for presentation or as part of other functionalities. We will not cover the IoT platform or the apps in this work. For more information on these aspects refer to [3].

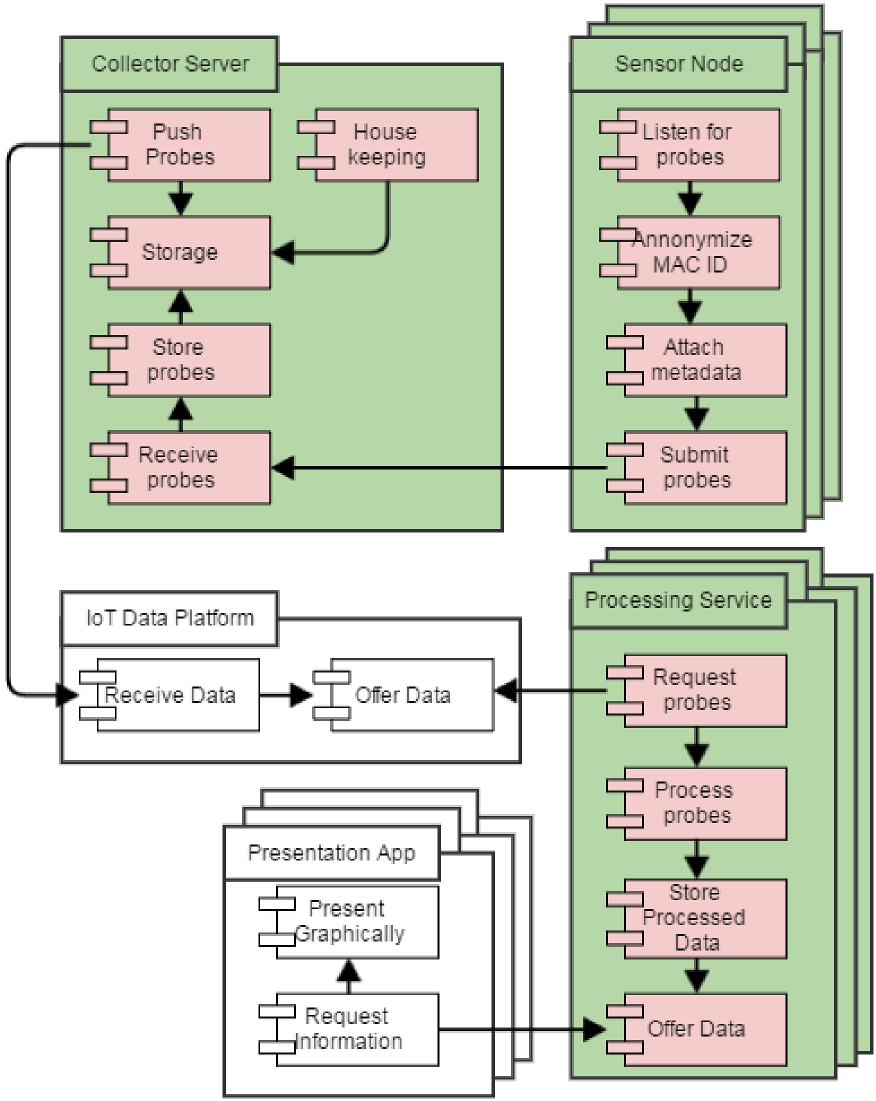


Fig. E.1: WiFi probe catching system architecture and module interactions (Updated from architecture diagram in [16]).

Sensor Node

The Sensor Node handles the following functionalities; probes scanning, anonymization, metadata collection, temporary storage, and submitting to Collector Server. As long as the Sensor Node is connected to a power source it is on and listening for WiFi probes. When the sensor receives a probe, the MAC address is anonymized by applying a hashing algorithm to a string composed of the date, a secret key, and the MAC address. This could potentially be reversible, but we go one step further and only save a substring of the resulting hash. This means that there is a chance of collisions, i.e. several MAC addresses could lead to the same hash. We see this as an acceptable trade-off in ensuring the privacy of the users. The anonymized ID is stored temporarily, along with location, timestamp and sensor ID, until it is scheduled to be transmitted to the Collector Server.

This means that for each probe received at the Sensor Node the following information will be collected:

- MAC address (anonymized)
- RSSI (the signal strength of the received probe)
- GPS location of the sensor
- timestamp from the sensor clock synchronized with GPS timestamps
- sensor ID (to distinguish probes from different sensors)

The Sensor Node is realized using a Raspberry Pi 2 model B as the main device. To this is attached a GPS dongle (u-blox 7 UBX-G6020), a WiFi dongle (TP-LINK TL WN722N), and a GSM module (Huawei E173) for back end communication. The Raspberry Pi runs Raspberian and the software components are written in Python and Java.

Collector Server

The Collector Server handles the following functionalities: Receiving and storing the collected probes, performing housekeeping regularly, and pushing the probes to the IoT platform. The receiving of the collected probes is done by the Sensor Node setting up a TCP connection secured using SSL. This is to ensure that only recognized sensors submit data to the server. As soon as the data is received it is stored in the local SQLlite database.

The housekeeping module will perform cleanup of old probes regularly, i.e. once every 24 hours. This is to ensure that the storage capacity of the Collector Server is not exceeded, but also because it is assumed that the collected probes will loose their relevance when they are old. Furthermore, as the collected probes are processed in the Processing Service to obtain the desired results it is assumed that this service will store the results for further backup, and not request old probes and redo the processing.

The collected probes are pushed to the IoT Platform which makes them available to the Processing Services, which can request probes based on location, sensor ID, time interval, or combine these.

Processing Service

There are many possible applications of collected WiFi probes, but in this work we consider people density estimation, and in particular, estimation of number of people on the bus. For this we will use the probes collected from one Sensor Node placed on a bus, and then perform some processing of the probes to give an estimate of the number devices and from that the number of people. For now we will assume the number of devices equals the number of people. We will update this assumption later.

E.3 Baseline Algorithm Definition and its Performance

In this section we first introduce the baseline algorithm for obtaining the estimated number of devices from the collected probes. Then we collect four sets of WiFi probes from a bus while manually observing the number of people. Next we evaluate the estimated number of devices from the baseline algorithm based on the impact of the $RSSI_{threshold}$ and the $Time_{threshold}$.

E.3.1 Definition of Baseline Algorithm

The WiFi probes can origin from devices on the bus and outside the bus. For this reason we will apply a simple algorithm to sort the probes from devices outside and on the bus. The result of this algorithm will serve as the base estimator which we will improve on in the rest of the paper.

The algorithm will use three types of information; the anonymized ID, the timestamp, and the RSSI value. The algorithm is based on the following assumptions:

- A bus passenger stays on the bus for some time and the probes from his/her device will be detectable for some time [16]
- Probes from devices on the bus will have higher RSSI than probes from devices outside the bus [16]

In the WiFi standard it is not defined how often a probe request should be sent from a device. This means that we cannot set a fixed time duration and expect probes from any present and active device to be captured within that. Actually the probe frequency varies greatly depending on the device state, the network chip, drivers, operating system, and others. However, according to Cisco CMX Analytics [8], probe request interval for smartphones is approximately once per minute when the device is in sleep mode (screen off) and approximately 10-15 times per minute when in standby mode (screen on). Furthermore, in [11] laboratory tests are done for several different brands of smartphones using different operating systems, which show large variability in probe frequencies. For this reason we will not introduce any assumption on probe frequencies of devices.

Based on the assumptions made about passenger traveling behavior and the behavior of RSSI of probes we now define a simple baseline algorithm to obtain the base estimate of the number of devices on the bus. The input to the baseline algorithm is probes submitted from the Sensor Node on the bus of interest, and specifically probes in a limited time interval. The probes are initially filtered according to the RSSI value, meaning that only probes with RSSI higher than a threshold $RSSI_{threshold}$ are used. It is assumed that the remaining probes now contain start and end times of devices on the bus. The travel time durations of the devices are evaluated according to a time threshold $Time_{threshold}$, and only devices that are present longer than this threshold are evaluated as being on the bus. This is summarized in the following:

```
list all device IDs
for each device ID
    list all probes with RSSI > RSSI_threshold
    for each RSSI filtered probe
        set as initial probe
        find time to all following probes for this ID
        if time to a probe > Time_threshold
            set device ID as in bus from initial probe to last
            break
```

Based on this algorithm, devices are registered as being in the bus between the first probe that fulfill the RSSI threshold requirement, and the last probe from this device.

E.3.2 Experimental Setup

We have performed four test runs with the Sensor Node on a city bus line. We did it on two different days and once in each direction per day.

While the sensor was collecting probes we manually counted the people entering and leaving the bus. We will use the result of this as the ground truth to evaluate the result of the baseline algorithm. As stated earlier we will initially assume that the number of devices equals the number of people.

The probes collected from the test runs are summarized in Table E.1 and plots related to the test runs can be seen in Figures E.2 to E.8.

From this we see that in the four data sets there are between 575 and 799 unique devices IDs. Furthermore it can be said that 94% of devices transmits 20 or less probes, while 59% of devices transmit between 2 and 20 probes.

The bus route goes from the university campus, through the city center

	Route from Campus		Route from City Center	
Date	Probes	Unique IDs	Probes	Unique IDs
27/4/16	5302*	575*	5296	798
10/5/16	5760	618	5359†	784†

Table E.1: Collected probes per test run (*=Test 1 and †= Test 2, selected for detailed analysis) [16].

and urban residential areas. This means that the route is very inhomogeneous, going through areas with high and low presence of people and devices.

We have chosen to focus on two of the datasets for detailed analysis (Test 1 and Test 2), which are indicated in Table E.1. We chose these two datasets because they show slightly different behavior. Figure E.2 presents both datasets in terms of unfiltered detected device presence (only probes with IDs that appear only once are removed) and the manually counted ground truth.

In Test 1 (see Figure E.2 top) the number of present devices lies around 20, with some variations in the beginning and end of the sequence, while the ground truth grows steadily from the beginning to the end. In Test 2 (see Figure E.2 bottom) the bus was less crowded with a maximum of 30 people about halfway through. The detected presence of devices follows the ground truth quite nice in the beginning, while increasing dramatically in the second half of the trip. For Test 2 the detected device presence fits well with the areas that the bus is passing through, i.e. in the city center with the train station and a lot of bus stops, there are many people and thereby many devices. For Test 1 we don not see the same clear increase in device presence when passing through the city center. This can be explained by the time of day of when the probes were collected, which was different from Test 2.

E.3.3 Impact of Device Presence Time Threshold

Now we will analyze the collected probes and in particular the duration of device presences. The goal of this is to evaluate the impact of a time threshold value to be used in the filtering of the probes. In Figure E.3, the device presence is plotted for the Test 1 and Test 2 datasets. Note that only devices that are present for at least 30 seconds are shown in the plot to make it easier to read.

Analyzing the device presence durations it is found that about 47% of the detected devices have a presence duration of less than 30 seconds.

These are most likely devices that are outside the bus, and therefor should be filtered away.

To evaluate the impact of the time threshold of different magnitudes, we evaluate probe datasets where we filter away probes from devices that are

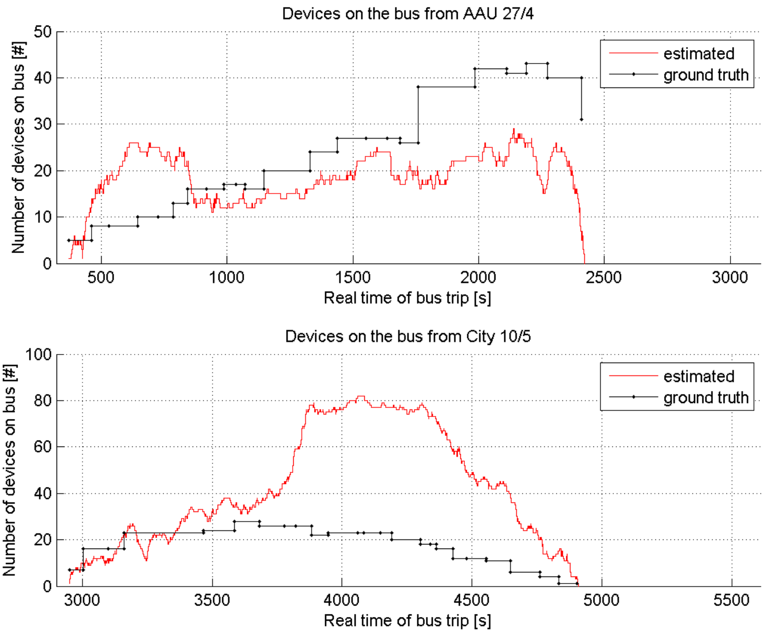


Fig. E.2: Presence of all detected devices aggregated as initial estimated number of people along with the manually counted ground truth [16].

present less than the time threshold. This can be seen in Figure E.4 where the time threshold is set to different values, and the resulting devices presence after filtering is plotted and compared with the ground truth.

From the graphs in Figure E.4 it can be seen that the time threshold only has limited effect. Some peaks disappear as the time threshold value increases but generally the graphs do not change significantly. More of the big peaks are filtered away as the time threshold is increased. This makes sense as the big peaks are present in areas with many pedestrians. But for Test 2 the part of the graph that highly overestimate the number of devices is not removed. This could be due to dense traffic or slow driving speed causing the sensor to collect probes from pedestrians and vehicles around the bus for longer durations. This supports the choice from earlier to also use the RSSI values in the baseline algorithm.

E.3.4 Impact of RSSI threshold

Now we will analyze the RSSI values of the collected probes to evaluate the impact of a RSSI threshold value to be used in the baseline algorithm. In

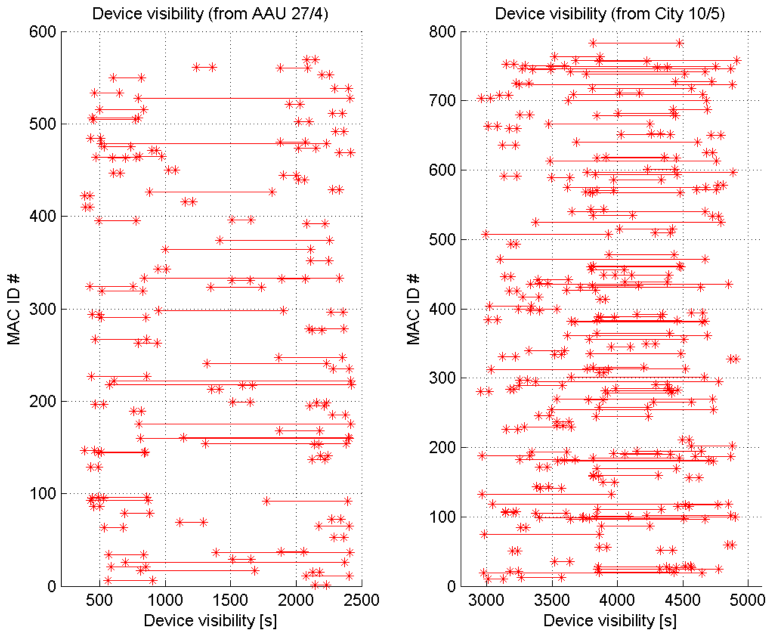


Fig. E.3: Device visibility duration (minimum time 30s) [16].

Figure E.5 the RSSI values of the collected probes in the two datasets can be seen.

The histograms have two noticeable peaks around -55dB and -85dB. We hypothesize that this is caused from devices on the bus and from devices outside the bus respectively. So setting a RSSI threshold at e.g. -60dB could potentially divide the collected probes as desired.

We now investigate the impact of the RSSI threshold of different values on the probe filtering. We do this by extending the filtering presented in Figure E.4 with different values for a RSSI threshold. We will now only consider a probe to be the first probe from a device if the RSSI value is higher than the RSSI threshold value. The result of this is presented in Figures E.6 and E.7.

The impact of the RSSI threshold value can clearly be seen when comparing Figures E.6 and E.7. In Figure E.7, for RSSI threshold of -65dB, the curves for the estimated number of devices now lie under or approximately follow the ground truth curves.

If we assume the percentage of passengers that carry an active WiFi enabled device is fixed, the ratio between the estimated number of devices and the number of passengers should stay approximately constant. In Figure E.8

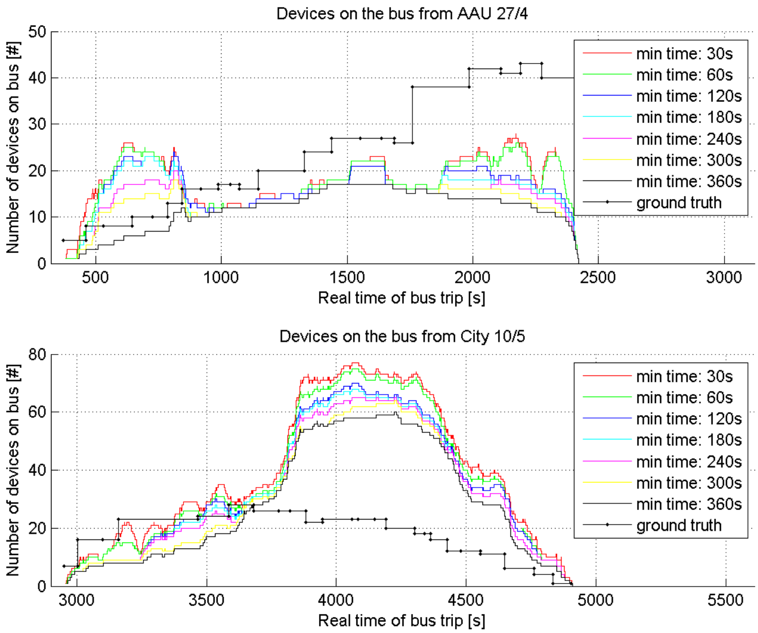


Fig. E.4: Number of devices filtered with different time threshold values [16].

we show the ratio between the ground truth and the estimated number of devices. Particularly the combination of time threshold at 360seconds and RSSI threshold at -65dB confirms the assumption as the ratio is approximately fixed around 50% with few variations.

E.3.5 Considerations of Baseline Algorithm Results

We have now evaluated the baseline algorithm and the estimated number of devices that it provides based on evaluation of the RSSI and Time threshold parameters. This approach is a low complexity approach of obtaining estimated number of devices on the bus. But as it was also indicated in the graphs, it lacks some accuracy in the obtained estimate. This is due to it estimating the number of devices, and not people. But when assuming that the number of devices equals number of person, i.e. assuming that the number of devices per person is constant, it is evident that the estimate graphs are inaccurate.

For this reason we would like to improve the estimate obtained from on the baseline algorithm. We will call the estimate from the baseline algorithm the base estimate, which we will improve on. Furthermore, we also want to

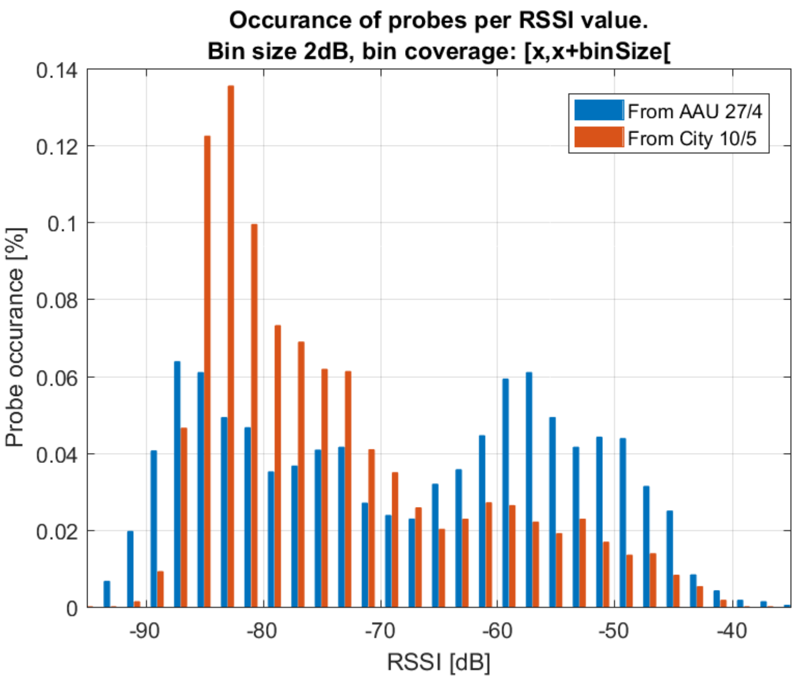


Fig. E.5: RSSI Histogram for Test 1 (27/4) and Test 2 (10/5) datasets.

explore the link between number of people and number of devices.

The reason why we want to improve on this approach and not develop a completely new approach, is that the complexity is low and we want to keep it like that. Having a low complexity estimation means that it is easier to deploy and tune according to the local area it is deployed in.

Furthermore, to make the estimate more useful in various use cases we also want to include an indication of the accuracy of the estimate. This will allow services and apps to evaluate how well it can trust the estimate.

E.4 Enchanced bus occupancy estimators

The final goal of the system is to be able to estimate the number of people on the bus, and to give an indication of quality of the estimate. The number of people on the bus is extrapolated from the number of WiFi enabled devices on the bus, which will be estimated and for that reason is subject to a number of errors in estimation. The number of WiFi enabled devices on the bus is estimated based on WiFi probes caught by the WiFi probe sensor on the bus. For each probe we collect timestamp (of sensor), GPS location (of sensor),

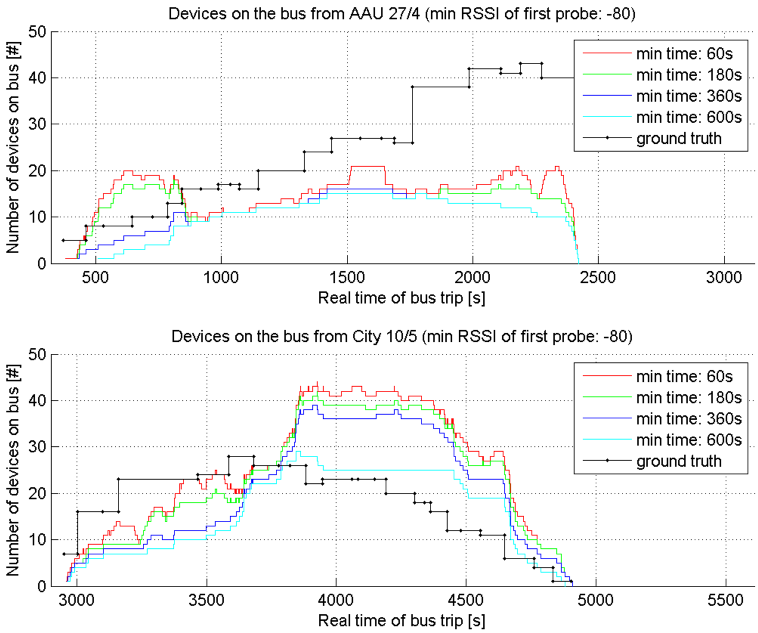


Fig. E.6: Number of devices filtered with different minimum times and minimum -80dB [16].

RSSI, and MAC address. After receiving the probes they are processed by the previously described baseline algorithm, taking into account the MAC address, timestamp and RSSI. The output of the baseline algorithm is the base estimate of number of devices. In the following we present a method to improve on the base estimator and show how the quality of the estimate can be derived.

E.4.1 Estimated Number of Devices

To improve the estimate we will now model it to understand the different components influencing the estimate. The estimated number of devices can be expressed like this:

$$E = D + FP - FN \tag{E.1}$$

Where:

E: Estimated number of devices on the bus
D: True number of devices on the bus
FP: (False positives) Number of devices falsely marked as on the bus

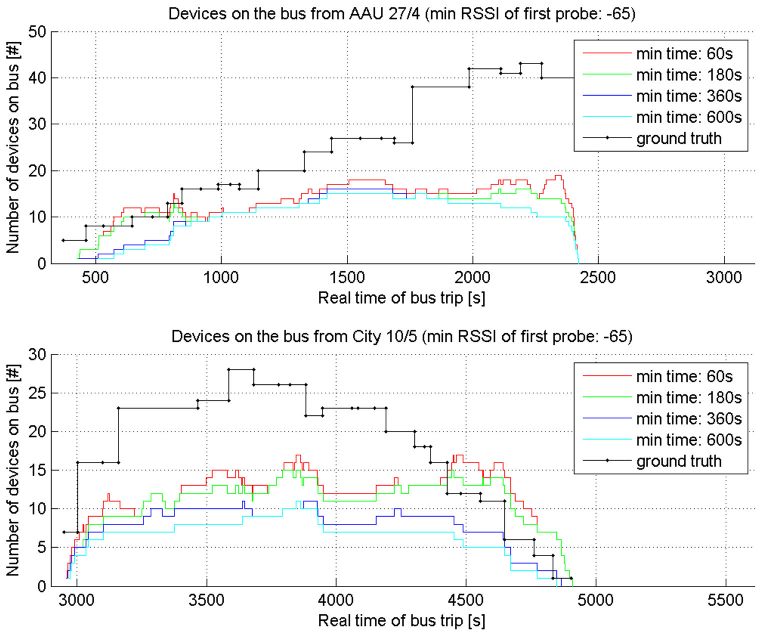


Fig. E.7: Number of devices filtered with different minimum times and minimum -65dB [16].

FN: (False negatives) Number of devices falsely marked as not on the bus

The estimated number of devices on the bus is the output of the baseline algorithm. This means that the mechanics of the baseline algorithm will also play some role in the accuracy of the estimation. These mechanics will depend on the parameters of the baseline algorithm, which are assumed fixed. The influence of these parameters on false positives and false negatives is described in the following.

E.4.2 False Positives

False positives (FP) occur in the situations when devices are estimated being on the bus, while this is not the case. The main factors causing FP are the following:

- Devices on the roadside outside the bus, but with a high enough RSSI, and visible long enough, to not be filtered out. This can be caused by the bus moving slowly through an area, e.g. due to dense traffic or traffic lights.
- Devices in other cars or on bikes driving next to the bus. These will

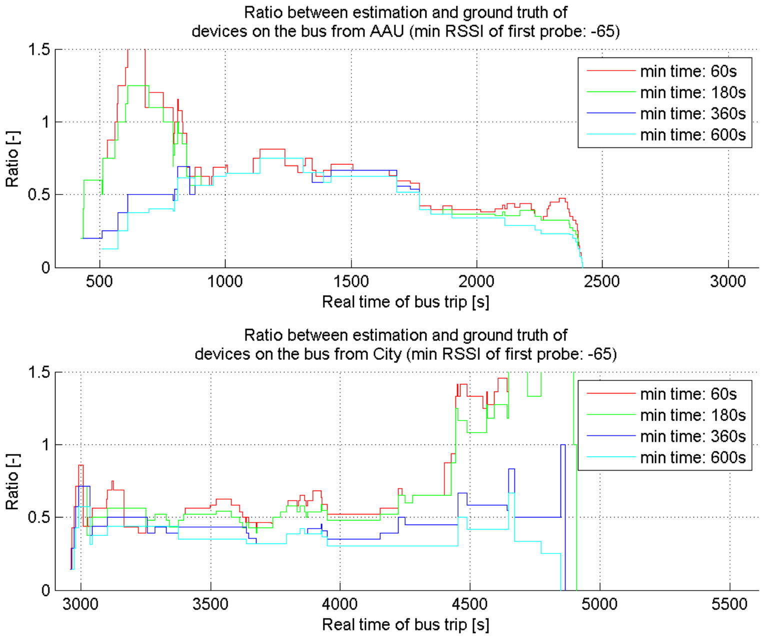


Fig. E.8: Ratio between estimated number of devices and ground truth with different minimum times and minimum -65dB for 27/4 and 10/5 [16].

look a lot like devices on the bus, provided the RSSI is not too low.

Assuming that each device outside the bus can be independently from other devices and with the same probability P_{FP} be wrongly estimated a being on the bus, FP can be modelled as a binomial random variable. Parameters of this distribution are $B(K, P_{FP})$, where K is the number of devices outside the bus in the area the bus is passing through, and P_{FP} is the probability of the individual devices being wrongly estimated as present on the bus.

E.4.3 False Negatives

False negatives (FN) arise in a situation when devices are one the bus but are not counted as on. FN depends on D, the number of devices on the bus. The main factors causing FN are the following:

- If a device is present on the bus for a duration too short to be accepted by the baseline algorithm, it will not be marked as on the bus.

- If the WiFi probe emission frequency of the device is too low, it is possible that the sensor will not catch two probes within the visibility window. Or maybe the device leaves the bus before a second probe is emitted.
- If WiFi probes are emitted from two devices simultaneously they might collide upon reception at the sensor, making the sensor unable to register one or both probes.
- If the probe catching sensor listens on the wrong channel, WiFi probes from devices will not be registered.
- If the devices employs MAC randomization then probes can be registered as originating from different devices, and the device will not be marked as on the bus.

Using the same argumentation as for FP we model FN to be binomial distributed, i.e. $B(D, P_{FN})$, where D is the true number of devices on the bus, and P_{FN} is the probability of the individual devices being estimated as not being on the bus.

E.4.4 Improved Estimator

The base estimator can be improved by using the relationship between E, D, FN and FP presented in Equation E.1. Note that Equation E.1 will hold if we apply expectation operation to both sides of the equation. Taking into account that FN and FP are binomially distributed, we obtain:

The expected value of the number of devices $E(D)$ is itself an estimator of the number of devices present on the bus. It presents an improved estimator compared to E, since it takes into account FNs and FPs. We denote this estimator as E_2. The detailed derivations of E_2 are presented in Equation E.2.

$$\begin{aligned} E\left(E\right) &= E\left(D\right) + E\left(B\left(K, P_{FP}\right)\right) - E\left(B\left(D, P_{FN}\right)\right) \\ E\left(E\right) &= E\left(D\right) + P_{FP} \cdot K - P_{FN} \cdot E\left(D\right) \end{aligned} \tag{E.2}$$

Finally, Equation E.2 can be rewritten as:

$$E\left(D\right) = \frac{E\left(E\right) - P_{FP} \cdot K}{1 - P_{FN}} \tag{E.3}$$

Motivated by Equation E.3, we now define a new estimator E_2 for the number of devices on a bus as follows:

$$E_2 := \frac{E - P_{FP} \cdot K}{1 - P_{FN}} \tag{E.4}$$

As can be seen from the equation, P_{FP}, P_{FN}, and K have a direct impact on E_2. When E estimates a high amount of devices on the bus but there are few people outside the bus, e.g. $E = 40$ and $K = 5$, E_2 gives a higher estimate than E as there can only be 5 FP but up to 40 FN. Similarly for the reverse case, e.g. $E = 10$ and $K = 40$, E_2 gives a lower estimate than E as the high K value causes there to be more FP.

E.4.5 Maximum Likelihood Estimator (MLE)

Another way of improving the base estimator is by applying Maximum Likelihood Estimation method. In the considered case no closed-form solution to the maximisation problem can be derived and we obtain MLE numerically.

The first step is calculating the joint probability distribution of D and E, $Pr(D = i, E = \alpha)$. This is done in Equation E.5.

$$
\begin{aligned}
&P_{DE}(i,\alpha) = \\
&Pr(D = i, E = \alpha) = \\
&Pr(D = i, \alpha = i + FP - FN) = \\
&\sum_{l=max(0,\alpha-i)}^{l=K} Pr(D = i, FP = l, FN = l - \alpha + i) = \\
&\sum_{l=max(0,\alpha-i)}^{l=K} Pr(D = i, FP = l, FN = j)
\end{aligned}
\tag{E.5}
$$

Where:

$$
\begin{aligned}
&j = l - \alpha + i \\
&max(0, \alpha - i) \leq l \leq K \\
&0 \leq \alpha \leq max(D) + K \\
&0 \leq i \leq D
\end{aligned}
$$

From the physical settings of the system and definitions of FN and FP, it is logically to assume that FP are independent of D and FN, while D and FN are dependent random variables. Independency assumption allows us to represent the joint probability as a product of individual probabilities. Furthermore, the expression for the joint probability of D and FN can be rewritten using definition of conditional probability. The resulting calculations are presented below.

$$
\begin{aligned}
&P_{DE}(i,\alpha) = \\
&\sum_{l=max(0,\alpha-i)}^{l=K} Pr(D=i, FP=l, FN=j) \\
&\sum_{l=max(0,\alpha-i)}^{l=K} Pr(FP=l) \cdot Pr(D=i, FN=j) \\
&\sum_{l=max(0,\alpha-i)}^{l=K} Pr(FP=l) \cdot Pr(D=i) \cdot Pr(FN=j|D=i)
\end{aligned}
\tag{E.6}
$$

Now using the assumptions of FP and FN being binomially distributed, and D being uniformly distributed between the bounds a and b, we obtain the following expression:

$$
\begin{aligned}
&P_{DE}(i,\alpha) = \\
&\sum_{l=max(0,\alpha-i)}^{l=K} \frac{1}{b-a} \cdot I(a \leq i < b) \cdot \binom{K}{l} P_{FP}^{l}(1-P_{FP})^{K-l} \\
&\qquad \cdot \binom{i}{j} P_{FN}^{j}(1-P_{FN})^{i-j}
\end{aligned}
\tag{E.7}
$$

where $I(a \leq i < b)$ is an indicator function that is equal to one if i is lying within the interval $[a,b)$ and it is zero otherwise.

We can now numerically calculate entries of the joint probability matrix of D and E, $P_{DE}(i,\alpha)$. The columns will show the probabilities for different values of D when observing a certain E. To limit the scope of the joint probability matrix we will delimit the parameters as follows:

- D is in the range $0 \leq D \leq 50$ (a and b from Equation E.7)
- E is in the range $0 \leq E \leq max(D) + K$
- $j = l - \alpha + i$

Figure E.9 illustrates the probability of D given that the base estimator is measured to be a certain fixed value. Calculations are done using Equation E.7. The example is given for three values of $\alpha = \{7, 15, 30\}$.

In general, after E is estimated to be a certain value α, the individual probability distribution for a number of devices D can be found by normalising a corresponding column in the joint pmf matrix, or writing formally:

$$
P(D=i|E=\alpha) = \frac{P_{DE}(i,\alpha)}{\sum_{i=0}^{i=50} P_{DE}(i,\alpha)} \tag{E.8}
$$

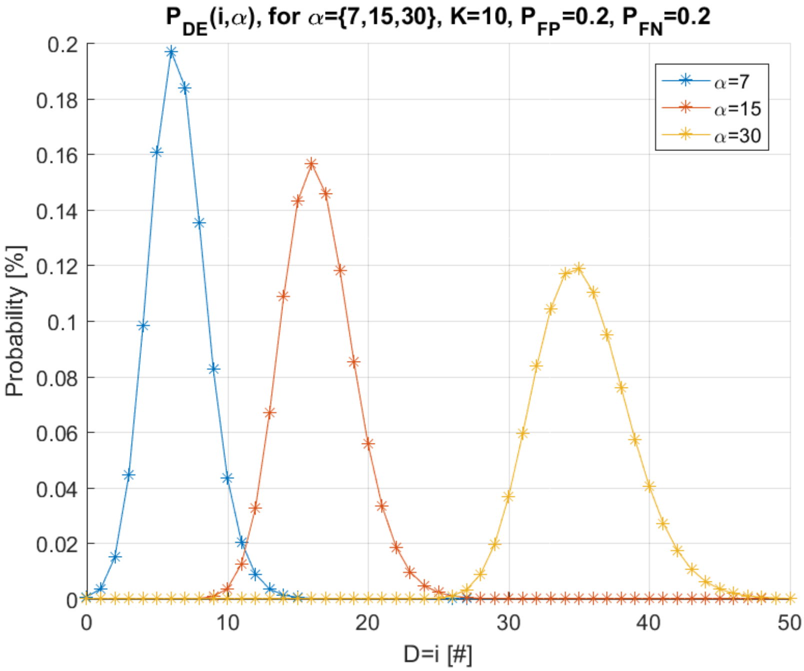


Fig. E.9: Probability of D given fixed E.

In our case the maximum likelihood estimator (MLE) is defined as follows

$$MLE = \arg\max_{i} P(D|E) \tag{E.9}$$

To continue on the example from Figure E.9 we calculate MLE for $\alpha = 15$. The result of this is shown in Figure E.10, where we also indicate the 70% credible interval. From this we see that $MLE = 16$, while the upper and lower bounds of the 70% credible interval are 13 and 19 respectively. Note that the 70% credible interval is found by eliminating 15% values on the outer right and left side of the graph on Figure E.10.

E.5 Finding Parameters

In the previous sections we described the two estimators E_2 and MLE, which improve on the base estimator. Furthermore, it was highlighted that in order to estimate the number of devices on the bus parameters K, P_{FN} and P_{FP} should be determined.

In this section we show how numerical values for these parameters can

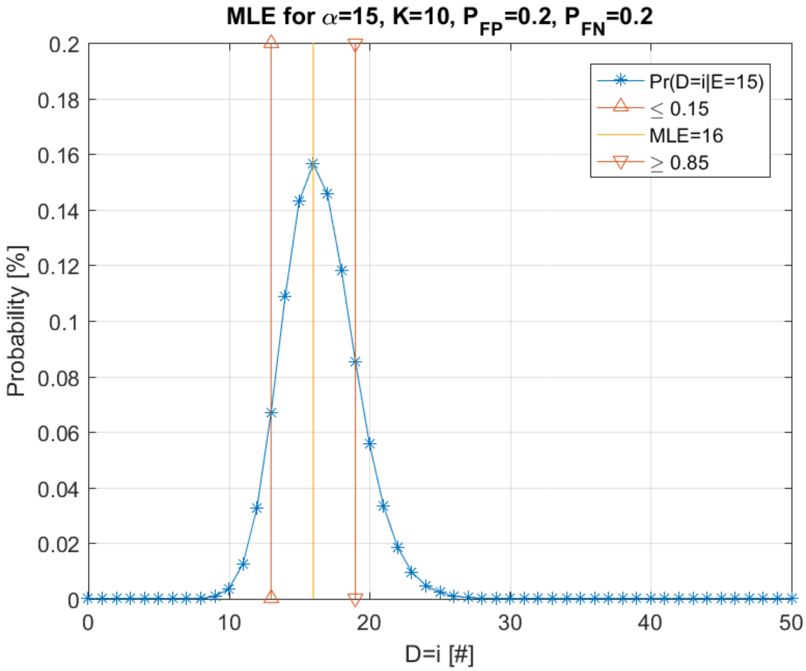


Fig. E.10: MLE when $\alpha = 15$ and the indication of 70% credible interval.

be found experimentally. These setups and the methodology we applied to obtain the parameters are described in the following.

Obtaining P_{FN}

We will obtain P_{FN} based on the knowledge of D and estimation of E. In case a precise number of devices on the bus, D , is known, and given that there are no devices outside the bus, all devices that are not detected by the algorithm from Section III should be counted as false negatives. Based on this observation, we construct the following measurement scenario. We use only known devices for the measurement and the number of devices is fixed throughout the whole experiment. Ideally, the measurements should be done on a bus in order to preserve signal propagation characteristics. From the collected probes, only probes that have a MAC address from the known set of devices will be used; other devices are filtered out. This is done to eliminate the influence of the devices outside the bus on the estimator. By this we get that K=0.

By taking the collected probes and applying the baseline algorithm, we will obtain E, the estimated number of devices on the bus. We can now evaluate true positives (TP) and false negatives (FN) based on E and D.

$$\begin{aligned} TP &= E \\ FN &= D - E \end{aligned} \tag{E.10}$$

Knowing D and having measured FN, we can derive P_{FN} as the ratio $\frac{FN}{D}$.

Due to variations in probe frequencies from different devices, and lost or missed probes, E could possibly vary over time. For this reason we will sample E with a sample step of 10 seconds, and from each sample of E derive P_{FN}, as described above. Finally we will average the sampled values of P_{FN} to obtain the final P_{FN} needed for our purpose.

Below we describe the results of our measurement campaign. The measurements were performed in a room, and not in a bus, for simplicity. The room however had similar propagation characteristics compared to a bus. There were a number of people, seats/chairs, similar form rectangular shape and size, and similar placement of the sensor. Devices and people were stationary during the measurement, much like passengers while the bus is driving. The measurement setup was as follows:

- 7 devices (D); LG Nexus 5X, Motorola Nexus 6, 3x Samsung Galaxy Y, 2x Samsung Galaxy S
- Not connected to a WiFi network
- Screen on (set to not turn off)
- WiFi sleep policy: never
- Measurement duration = 40min
- Minimum of 4 WiFi networks saved on each device

Figure E.11 shows when probes are received from the different devices over time. From this figure it is evident that different devices will emit probes at different rates, meaning that the time threshold in the baseline algorithm will have a great impact.

Figure E.12 shows histograms of the signal strength of the collected probes. This is to get an idea of the signal strength distribution, to understand the impact of the RSSI threshold. From this figure it is evident that the signal strength can vary noticeably despite both sensor and devices being fixed in location.

Now we will derive FN based on the probes collected in this scenario. We do this as stated in Equation E.10, by subtracting the base estimator E from the known number of devices D. One should note that the estimator E is smaller than D in two cases:

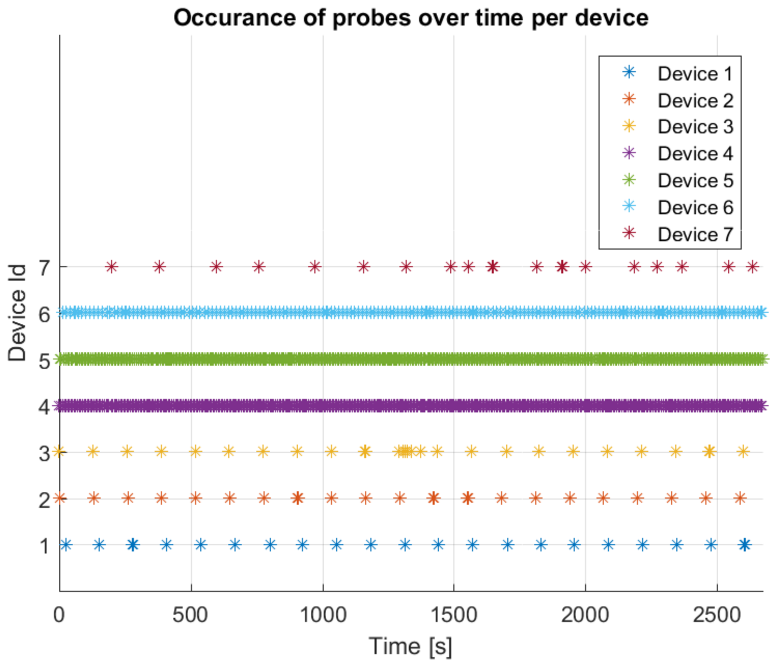


Fig. E.11: Probes received for each individual device. This gives an overview of probe frequency behavior.

- when a signal strength of a received signal is low and it is filtered out by the algorithm as not being present on the bus, but being outside the bus;
- when a device is present on a bus only for a short time period, and the algorithm wrongly concludes that the bus is just passing a device outside the bus.

To get an overview of what impact the signal strength of the received signal has on P_{FN} we obtain FN and calculate P_{FN} for a wide rage of RSSI threshold values. Different instances of the algorithm are run varying RSSI threshold in the range $\{-94db : -24db\}$ in steps of 1 db. In order to account for the second reason for wrong estimation, we emulate time a device is present on the bus by varying observation time from 10s to 600s. One should note that if we observe for sufficient long time, all devices on the bus will be eventually detected. Looking at FigureE.11 it is clear that the detection probability of a device after observation period of 600s is nearly 100%, even if some probes are lost. Figure E.13 summarises the results. One can observe that at 600s P_{FN} is only due to the weak signals and/ or high RSSI threshold

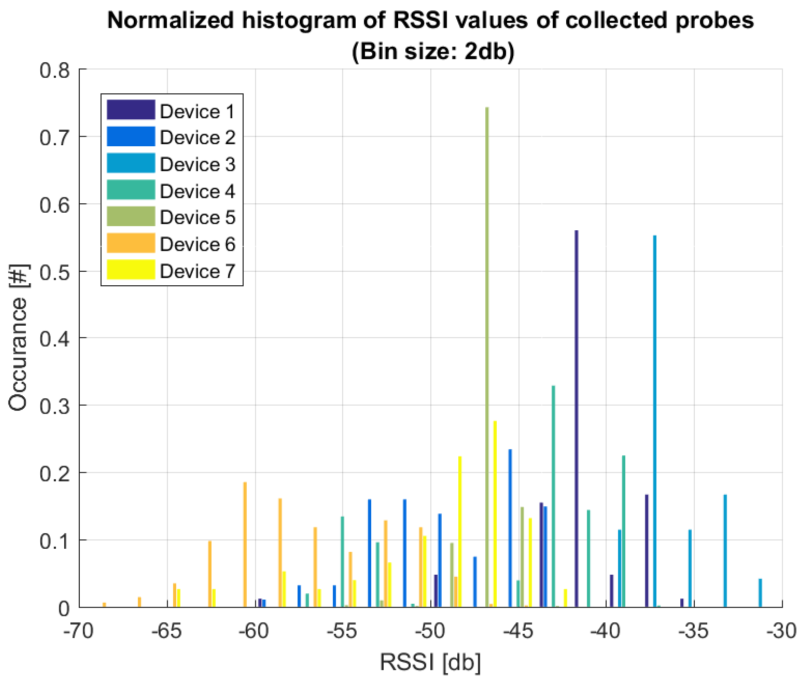


Fig. E.12: Graph of the RSSI values per device, showing that even though location fixed the RSSI can vary.

set in the algorithm.

It is reasonable to assume that a person is present on a bus at least for a duration of time between two bus stops. For the bus stops in Aalborg inner city this duration is about 2-3 min. From Figure E.13 we can see that starting from approximately 200s and on the duration of the observation time almost does not influence values for P_{FN}. For this reasons in our future considerations we disregard the observation time and consider P_{FN} only as a function of RSSI threshold calculated for fixed observation time of 250s.

Obtaining K and P_{FP}

P_{FP} can be obtained based on the knowledge of FP and K. FP can be measured directly when driving in a bus for which no activated WLAN devices are on board (D=0). In that case, the output of E are automatically all false positive counts.

Practically we collect probes to base K on, by putting the sensor in a car, only with the driver, and drive along the bus route. We will be driving the same speed as the bus, however not near any bus. This will mimic the

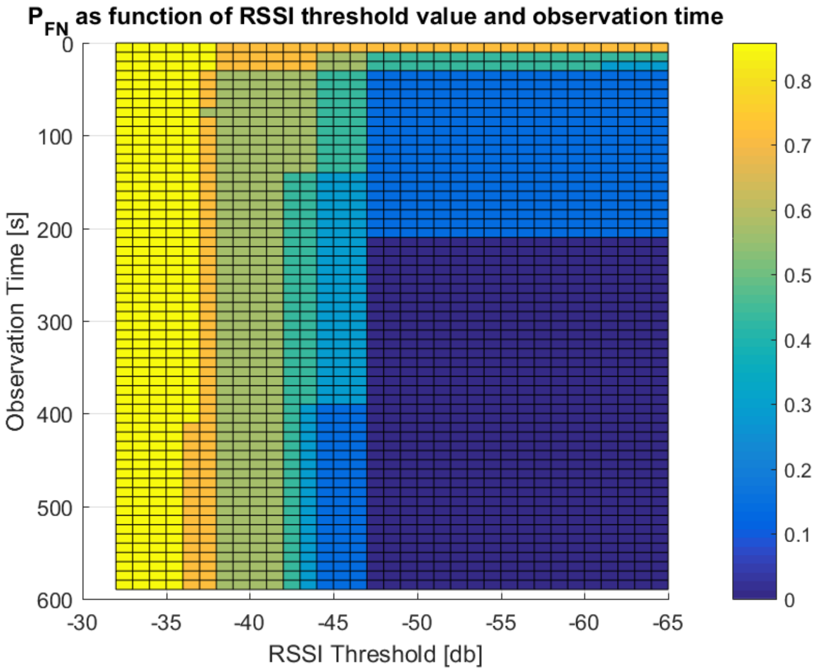


Fig. E.13: P_{FN} based on FN, with varying RSSI thresholds and at different observation times.

sensor being on an empty bus while driving the bus route, and only collecting probes from devices outside the bus. We assume the channel characteristics are similar for the path between the sensor and the devices when the sensor is in a car compared to being in a bus.

K is defined as the number of devices outside the bus, that the sensor can collect probes from, meaning the number of unique MAC addresses of the collected probes equals K. Probes are emitted from devices with varying frequency depending on the device model and state, which means that K can not be determined instantaneously, but must be estimated over a time duration.

The bus that carries the sensor will drive with different speeds, why the area covered within a fixed duration can vary, i.e. driving 50km/h or 30km/h for a duration of 30 seconds equals 416m and 250m respectively. This means that the sensor will cover a greater area if the speed is higher, but at the same time probability to receive a probe from a single device is lower since the bus passes quickly through the device communication range zone.

Ideally we would like K to be approximately the same size for a specific type of area depending only on the density of population outside the bus, no matter the speed of the sensor. Therefore, in the following K is assumed to be constant within a certain area type (rural, urban, city center). We adjust the size of the time interval for observing probes according to the speed of the bus, practically meaning that we want to keep the size of the area covered approximately constant. We base the size of this area on the WiFi range, which we assume to be 150m in an outdoor environment with few obstacles. Based on this we define the area to be a circle with a radius of 150m. With a bus speed at 30km/h or 50km/h the sensor will have a fixed device within its communication range for 36s and 21.6s respectively, assuming that the bus drives in a straight line passing right by the device. Here we are not considering mobile devices moving along or opposite direction of the bus.

This is the basis we will use for deciding the duration of the time window within which we will count K as the number of unique MAC addresses from devices. Based on the timestamps and GPS locations in the collected probes the time windows are ranging from 22 seconds to 121 seconds, with an average size of 52.78 seconds.

Finally we will apply the baseline algorithm on the collected probes, in the same way as described in previous section. This will return E as a discrete step function. But in this case E equals FP, as we have D=0, i.e. no devices on the bus, and thereby also FN=0. From FP, by assuming it being Binomial distributed ($FP \sim binom(K, P_{FP})$), we can derive the parameter P_{FP}.

The measurement data estimating K were collected while driving in a car along the bus route shown on Figure E.14. In this measurement scenario we collected 4950 WiFi probes over a duration of 34 minutes. On the figure its also indicated how the probes are divided according to area type. Three area types are defined: Sub urban, urban and city center. We used our knowledge of Aalborg area and surroundings in order to make this division. As we will see later, this division is also supported by measured different values of K.

Evaluating the RSSI values of the collected probes we find that 95% of the probes have RSSI values lower than -65dB. This is a clear contrast to the probes presented in Figure E.12 while estimating P_{FN}.

The whole data set collected for K is divided into a number of sampling subsets following the rule: a subset is constructed by adding probes received within 30 seconds. Afterwards it is checked if a minimum distance of 150 meters is covered. If so the subset is complete and the next is made. Otherwise the subset is extended until it will cover 150 meters or have a time duration of 120 seconds. From this measurement scenario and with the described subset division we obtain 37 subsets.

For each subset we obtain K as the number of unique devices, meaning the number of devices that we receive at least a single probe from. The K-values are plotted as a step function in Figure E.15. By applying the area type division (as shown in Figure E.14) we split the K-values and average them to obtain a single value per area type. The averaged values for K is indicated in

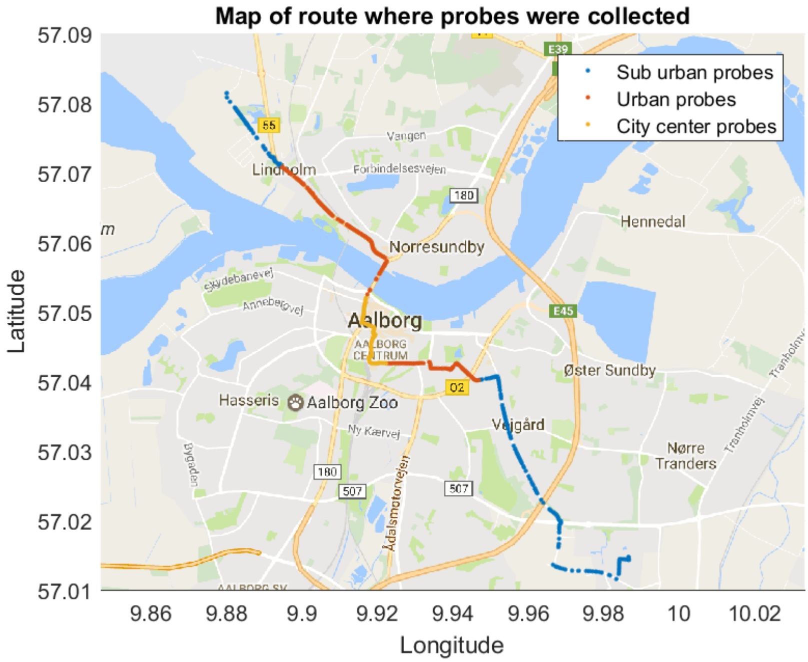


Fig. E.14: Map of the route driven while collecting probes.

the figure and they will be used later on in the estimation.

Next we obtain FP and from that calculate P_{FP}. FP is obtained by running the full dataset through the baseline algorithm. In this case both threshold parameters of the algorithm will have impact on the FP value. Different instances of the algorithm are run varying RSSI threshold parameter from -94dB to -24dB in steps of 1dB and varying time threshold in the range from 1s to 601s in steps of 10 seconds. The result is presented in Figure E.16. From this figure it can be seen how P_{FP} behaves under different threshold pairs. We will use this for selecting threshold pairs in the following.

E.5.1 Selection of Threshold Pairs

Based on the values for P_{FN} and P_{FP} presented in Figures E.13 and E.16 we now select a number of different values for RSSI and time thresholds, that we refer to as the threshold pairs. We select these pairs such that they represent different operating conditions. These pairs will be used in the evaluation of the improved estimator later. The eight selected pairs and corresponding P_{FN} and P_{FP} can be seen in Table II.

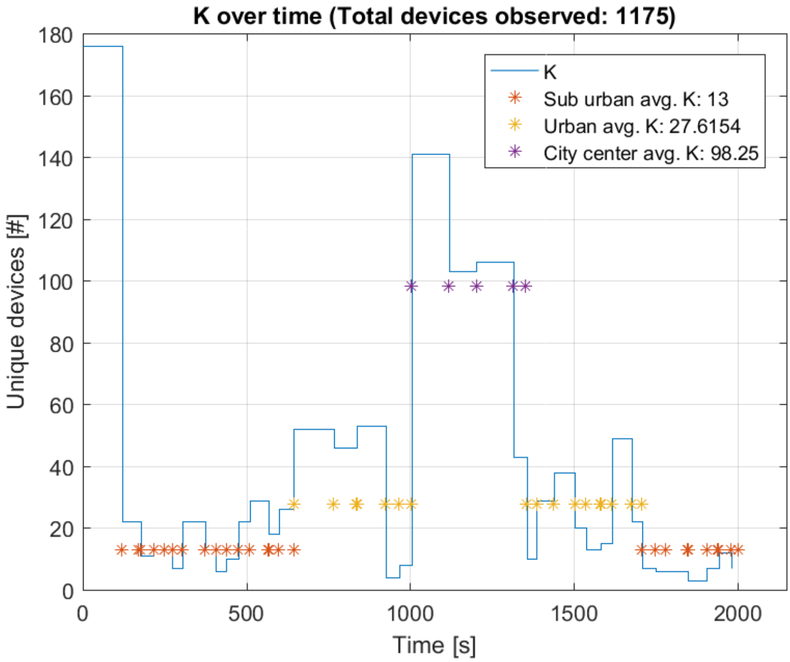


Fig. E.15: K (number of unique devices) observed over time, and average K per area type.

	$Thres_{RSSI}$	$Thres_{Time}$	P_{FN}	P_{FP}
TP_1	-92	31	0.0247	0.1191
TP_2	-89	541	0.0247	0.0306
TP_3	-87	171	0.0247	0.0698
TP_4	-66	261	0.0247	0.0102
TP_5	-51	180	0.0252	0
TP_6	-45	180	0.1837	0
TP_7	-40	180	0.5789	0
TP_8	-31	180	0.8604	0

Table E.2: Selected threshold pairs for $Thres_{RSSI}$ and $Thres_{Time}$, and the resulting P_{FN} and P_{FP} read from Figures E.13 and E.16.

E.6 Estimated Number of People

The final goal of this system is to be able to estimate the number of people on the bus. In the previous sections we described E as the base estimator for number of devices D. From E we could obtain E_2 as an improved estimator

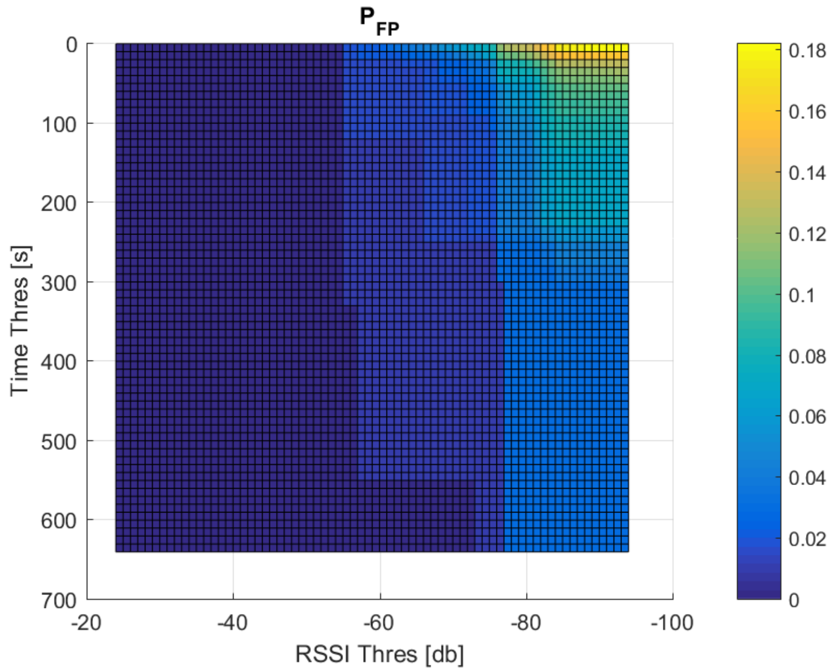


Fig. E.16: P_{FP} based on FP with varying time and RSSI thresholds.

of D. Now we need to make the final link between number of devices D and number of people that we denote as N.

Given that the probabilities of a person having 0, 1, 2, 3, etc. active WiFi enabled devices is known, and given the number of people on the bus, we can calculate the probability of a certain number of devices present on the bus, i.e. $D|P$. As a next step we apply maximum likelihood optimization to find the most likely number of people based on the estimated number of devices, N_{ML}.

In practice it is difficult to obtain the precise distribution of a number of devices per person. To approximate this distribution we use some recent statistics [17]. According to it, the number of mobile devices per capita in Denmark as of 2007 is approximately 1.14. We will conservatively assume that the number of devices per capita in Denmark at the time of measurement in 2016 is 1.2. Based on this we model the probability of a person having a number of devices as a discrete finite distribution with probabilities of a person having a number of devices defined as follows: $P_0 = 0.23, P_1 = 0.48, P_2 = 0.18, P_3 = 0.08, P_4 = 0.03]$ These probabilities mean that it is most likely that a person has 1 active WiFi enabled device. The probability drops significantly for 2, 3 and 4 devices. We assume that no one are carrying more than 4 devices. The respective probabilities are selected in a way that on average a person would carry 1.2 devices.

Figure E.17 provides an example of probability distribution for D given N calculated for 3 different values of N. Obtained distributions are based on 100000 realizations.

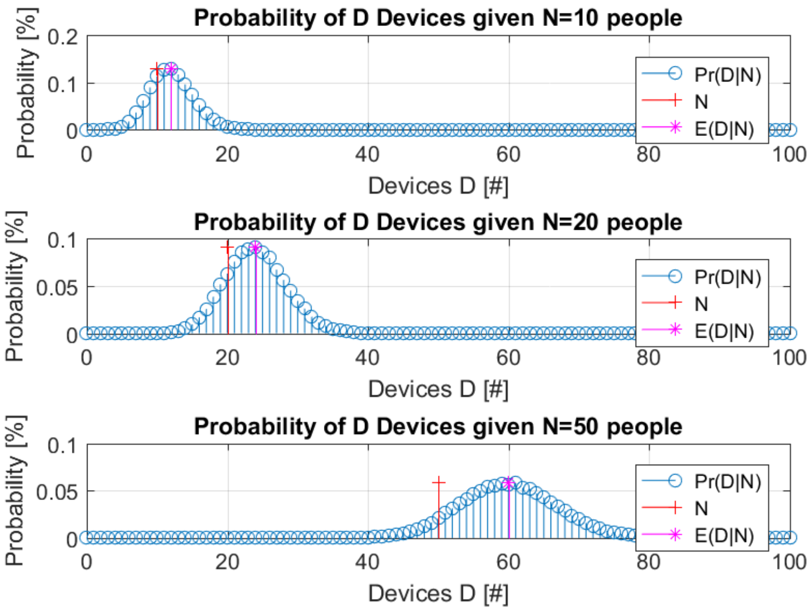


Fig. E.17: Plot of probability distribution of number of devices given number of people $D|N$ for 10, 30 and 50 people.

Maximum likelihood estimator of a number of people on the bus is obtained as

$$N_{ML} = \arg\max_k P(N = k|E) \tag{E.11}$$

Probability distribution for the number of people on the bus given the base estimator can be found using knowledge of conditional distribution of the number of people given the number of devices and distribution of the number of devices given the base estimator:

$$P(N = k|E = \alpha) = \sum_{i=0}^{i=d_{max}} P(N = k|D = i)P(D = i|E = \alpha) \tag{E.12}$$

where d_{max} is the maximum possible number of devices. The last term in the product in Equation E.12 we obtain from the joint probability P_{DE}. The first term can be found by using the following expression:

$$P(N = k|D = i) = P(D = i|N = k)P(N = k)/P(D = i) \tag{E.13}$$

In Section IV we have assumed that D is a uniformly distributed random variable. Now we also assume that N is uniformly distributed and both terms $P(N = k)$ and $P(D = i)$ are constant, and thus they do not influence maximization operation. Thus, Equation E.11 can be rewritten as

$$N_{ML} = \arg\max_{k} \sum_{i=0}^{i=d_{max}} P(D = i|N = k)P(D = i|E = \alpha) \tag{E.14}$$

In Section VII we use Equation E.14 to find MLE for the number of people on the bus. Note that one can obtain more precise estimator if precise distributions of N and D are known, if some statistic is available.

E.7 Results

We will now evaluate the improved estimator MLE in terms of different threshold value pairs presented earlier. The result will be a recommendation of which threshold values to use in which settings.

We will evaluate the estimator based on a dataset of probes collected on a bus. Furthermore, ten different geographical locations are selected as evaluation points. The bus route, and the evaluation points, are indicated on Figure E.18.

The evaluation points are distributed in the different area types as follows: There are 4 evaluation points in sub urban, 3 in urban and 3 in city center area type. While the probes were being collected a person was present on the bus counting the people getting on and off at bus stops. This will be denoted as the ground truth of people N_{GT}. Table E.3 presents the evaluation points and the corresponding values for K and N_{GT}.

In Figure E.19 the distributions of P_{DE} and $D|N$ are presented for EP_5 and TP_3. For calculating $P_{DE}(i, \alpha)$ we use the value of E at the evaluation point, and P_{FP} and P_{FN} which are based on the threshold values and the measurements presented in earlier sections. Furthermore, the K-value used is chosen based on the area type of the evaluation point as presented in Figure E.15. $D|N$ is calculated based on the multiple device modeling described in Section E.6. For both $P_{DE}(i, \alpha)$ and $D|N$ we have indicated the 70% credible interval of the probability distributions and also MLE and the expected value of $D|N$. Following the procedure described in Section VI we find that $N_{ML} = 15$ for this case.

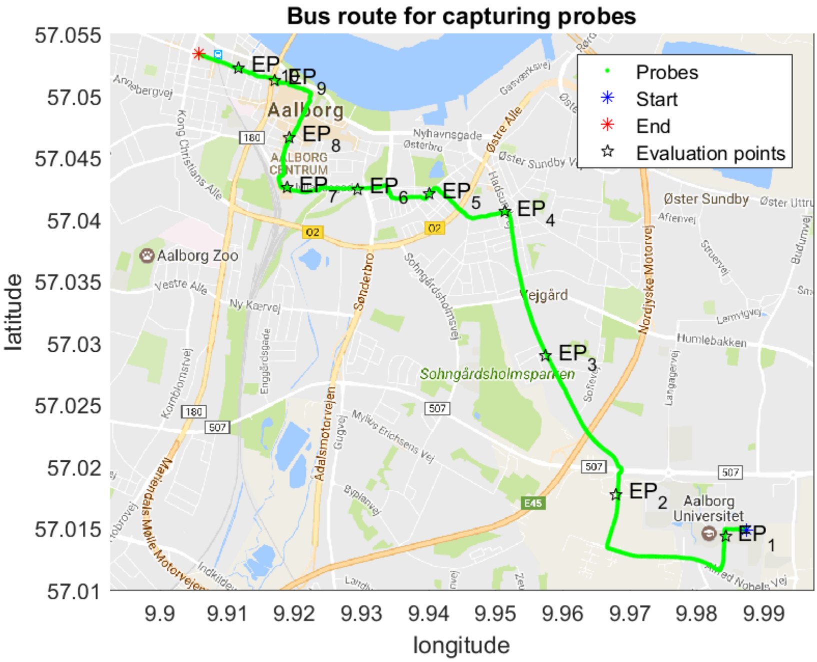


Fig. E.18: Plot of the bus route where probes were collected, including the start and end point. Also the evaluation points are indicated.

	Area type	K	N_{GT}
EP_1	Sub urban	13	5
EP_2	Sub urban	13	7
EP_3	Sub urban	13	8
EP_4	Sub urban	13	8
EP_5	Urban	27	10
EP_6	Urban	27	12
EP_7	City center	98	12
EP_8	City center	98	5
EP_9	City center	98	2
EP_{10}	Urban	27	3

Table E.3: The evaluation points where the approach is evaluated. K-values are chosen based on the measured K values presented on Figure E.15. Values for N_{GT} are observed on the bus.

In Table E.3 the N_{GT} values for the different evaluation points can be seen, and in Table E.4 and in Table E.5 the N_{ML} and N_E values for the different

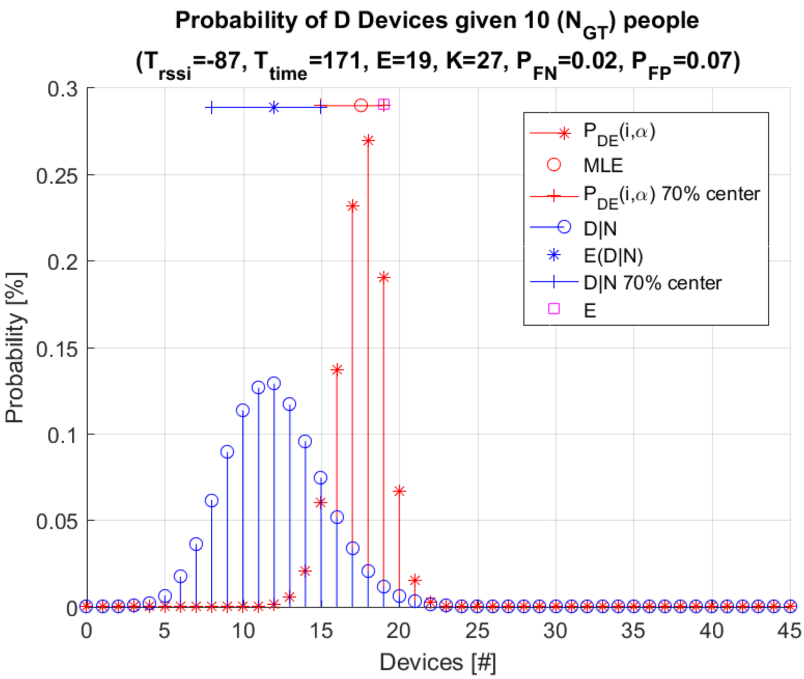


Fig. E.19: Distribution, expected value and 70% center interval of $P_{DE}(i, \alpha)$ and $D|P$ at EP_5.

combinations of threshold pairs and evaluation points are presented.

	TP_1	TP_2	TP_3	TP_4	TP_5	TP_6	TP_7	TP_8
EP_1	15	1	4	0	0	0	0	0
EP_2	9	6	7	6	5	3	4	6
EP_3	10	7	10	7	5	4	6	6
EP_4	17	13	15	13	10	9	14	6
EP_5	16	12	15	13	12	9	16	6
EP_6	14	10	12	12	11	8	14	6
EP_7	5	8	9	13	12	10	18	6
EP_8	4	6	6	10	9	8	14	0
EP_9	3	3	6	8	7	6	8	0
EP_{10}	8	4	8	7	5	4	6	0

Table E.4: The N_{ML} value per combination of evaluation point and threshold value pair.

To figure out which of the threshold value pairs perform best we calculate the Mean Squared Error (MSE) per threshold value pair. The MSE between

	TP_1	TP_2	TP_3	TP_4	TP_5	TP_6	TP_7	TP_8
N_E	19	1	5	0	0	0	0	0
N_E	12	7	9	7	6	3	2	1
N_E	13	9	12	8	6	4	3	1
N_E	22	15	19	15	12	9	7	1
N_E	22	15	19	15	14	9	8	1
N_E	20	13	16	14	13	8	7	1
N_E	17	12	17	16	14	10	9	1
N_E	16	10	14	13	11	8	7	0
N_E	15	7	14	10	8	6	4	0
N_E	13	5	11	8	6	4	3	0

Table E.5: The N_E value per combination of evaluation point and threshold value pair.

N_{GT} and N_{ML}, and N_{GT} and E, are shown in Table E.6. From thus we see

MSE	TP_1	TP_2	TP_3	TP_4	TP_5	TP_6	TP_7	TP_8
N_{ML}	30.5	7.0	13.0	13.9	9.2	10.5	27.6	16.0
N_E	106.5	14.6	55.2	27.2	14.8	10.5	12.2	52.0

Table E.6: MSE between N_{GT} and N_{ML}, and between N_{GT} and E.

that our estimated number of people N_{ML}, which is based on our improved estimator, generally achieves lower MES for most of the threshold pairs. Furthermore, using threshold pairs TP_2 or TP_5 gives us a good estimation.

The MSEs obtained for these threshold value pairs are relatively close, so we will take a closer look at the the implications of the threshold value magnitudes.

- TP_2: Low RSSI threshold and very high time threshold results in the situation where almost no probes are filtered based on the signal strength, but only the devices that are observed for a very long time are counted as being on the bus.

- TP_5: Medium RSSI threshold and medium time threshold guarantees that filtering done rejecting both weak signals and probes that are observed for less than 3 min.

One can also observe that the estimation using the threshold pair TP_2 is performing slightly better in suburban areas, while estimations in the city areas is less precise. It is logical as in the suburban areas there are fewer devices outside the bus, bus is driving faster and typically people are staying longer on the bus. Looking closer at the performance of the estimator using pair TP_5, the situation is the opposite. Here it shows the best performance in the city center. In the city it is necessary to filter devices with low RSSIs out as the device density outside the bus is high.

E.8 Design of Quality Indicator

A quality indicator of the estimated number of people can be now designed. The quality indicator is useful for services and applications using the estimated number of people, who can evaluate the quality of the information and then act accordingly.

The estimated number of people depend on the estimator for number of devices $P_{DE}(i, \alpha)$. But as the application is using the number of people we need to relate the quality indicator to the number of people and not the number of devices.

We will base the quality indicator on the range between upper U_P and lower L_P bounds of the 70% credible interval of the joint probability density between $P_{DE}(i, \alpha)$ and $D|N$. We will normalize the range according to N_{Max}, the maximum number of people that we can estimate, i.e. the number of people there is room for on the bus. The expression for the quality indicator QI is defined as follows.

$$QI = 1 - \frac{(U_P - L_P)}{N_{Max}} \tag{E.15}$$

In our scenario $N_{MAX} = 72$ for the bus that we collected probes on in Aalborg.

In Table E.7 we see the QI values for the 8 threshold pairs at the 10 evaluation points presented in Tables E.2 and E.3.

From Table E.7 we see that the quality indicator is mostly above 0.8, indicating that the ranges of the estimated number of people mostly are small. It is worth noting that in most of the cases the actual number of people on the bus is low, i.e. 12 or less people. For this reason the number of devices are also low, not allowing for bigger ranges of estimated number of people.

E.9 Summary and Outlook

In this work we have presented a maximum likelihood estimator (MLE) of number of devices on a bus. This estimator is an improvement of a baseline estimator, which is the output of a baseline algorithm. The baseline algorithm performs a simple filtering of WiFi probes collected via a WiFi probe collection system also presented here. The filtering is performed based on threshold values for device presence time and RSSI of the probes. From the

	TP_1	TP_2	TP_3	TP_4	TP_5	TP_6	TP_7	TP_8
EP_1	0.96	0.99	0.96	0.99	0.99	0.99	0.99	0.99
EP_2	0.96	0.97	0.96	0.99	0.99	0.97	0.94	0.92
EP_3	0.96	0.97	0.96	0.97	0.99	0.96	0.93	0.92
EP_4	0.96	0.96	0.94	0.97	0.97	0.96	0.90	0.92
EP_5	0.94	0.96	0.94	0.96	0.97	0.96	0.90	0.92
EP_6	0.93	0.96	0.94	0.96	0.97	0.94	0.90	0.92
EP_7	0.93	0.94	0.93	0.96	0.97	0.94	0.89	0.92
EP_8	0.94	0.94	0.93	0.96	0.97	0.94	0.90	0.99
EP_9	0.94	0.96	0.93	0.96	0.97	0.96	0.92	0.99
EP_{10}	0.94	0.97	0.94	0.97	0.99	0.96	0.93	0.99

Table E.7: QI Quality indicator value for different combinations of threshold value pairs and evaluation points presented in Tables E.2 and E.3.

MLE we obtained an estimate of number of people, based on a simple model for number of devices per person. In the end we designed a quality indicator for the estimator for number of people.

The goal in our case is to estimate the number of people on a bus, meaning that the range of possible number of people is limited. This leads to the design of the quality indicator which easily is scaled according to the size of the bus in terms of passengers.

The quality indicator will give applications and services using the information from the estimation a basis to evaluate how well they can trust the information. The quality indication can also be used in merging the number of people estimate with estimates from other sources. Knowing the quality of the estimate could for instance be used for assigning weighting to information before merging it.

Our approach to create the estimator is based on stochastic modeling, including measurements in experimental setups to estimate the needed parameters for the model. Another approach could be to apply machine learning on the collected probes. This would reduce the need for additional measurements for fine tuning the baseline algorithm and estimating model parameters.

In the baseline algorithm we consider the time stamp and RSSI attached to the probes. Another approach could be to utilize the GPS location attached to each probe. For instance evaluating when the bus is moving, and probes are received from the same device, the device is more likely to be on the bus than while the bus is not moving. This would also handle the possible issue of how to evaluate if a device is getting on or off the bus when the bus is stopped at a bus stop.

In the WiFi probe collection system the Sensor Node currently only collects probes on 1 channel, approximately covering one third of the 2.4 GHz spectrum. The impact of this should be explored, i.e. collect probes from the full 2.4 GHz spectrum to validate that we do not miss devices when only collecting probes from one part.

Another point that has not been covered in this work is MAC randomization, which is employed in recent version of both Android [1] and iOS [2] when devices perform background scanning for WiFi APs. This will most likely have an impact on the accuracy of the estimation performed in this work. The impact of this should be investigated to come up with solutions for how to handle it in the estimation.

Acknowledgments

This work is supported through the BIG IoT project (Bridging the Interoperability Gap of the Internet of Things) funded by the European Union's Horizon 2020 research and innovation programme under grant agreement No 688038.

References

[1] Android Developer, "Android 6.0 changes," February 2017. [Online]. Available: https://developer.android.com/about/versions/marshmallow/android-6.0-changes.html

[2] Apple Inc., "About the security content of ios 8," February 2017. [Online]. Available: https://support.apple.com/en-us/HT201395

[3] BIG IoT, "Big iot," February 2017. [Online]. Available: http://big-iot.eu/

[4] M. Bøgsted, R. L. Olsen, and H.-P. Schwefel, "Probabilistic models for access strategies to dynamic information elements," *Performance Evaluation*, vol. 67, no. 1, pp. 43 – 60, 2010. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0166531609001175

[5] T. Brade, S. Zug, and J. Kaiser, "Validity-based failure algebra for distributed sensor systems," in *2013 IEEE 32nd International Symposium on Reliable Distributed Systems*, Sept 2013, pp. 143–152.

[6] N. Caceres, J. P. Wideberg, and F. G. Benitez, "Deriving origin destination data from a mobile phone network," *IET Intelligent Transport Systems*, vol. 1, no. 1, pp. 15–26, March 2007.

[7] F. Calabrese, M. Colonna, P. Lovisolo, D. Parata, and C. Ratti, "Real-time urban monitoring using cell phones: A case study in rome," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 1, pp. 141–151, March 2011.

[8] Cisco, "Cmx location analytics," 2016. [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.html

[9] ——, "White paper: Cisco visual networking index: Global mobile data traffic forecast update, 2015-2020," February 2016. [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.html

[10] J. Figueiras, H. P. Schwefel, and I. Kovacs, "Accuracy and timing aspects of location information based on signal-strength measurements in bluetooth," in *2005 IEEE 16th International Symposium on Personal, Indoor and Mobile Radio Communications*, vol. 4, Sept 2005, pp. 2685–2690 Vol. 4.

[11] J. Freudiger, "How talkative is your mobile device?: An experimental study of wi-fi probe requests," in *Proceedings of the 8th ACM Conference on Security & Privacy in Wireless and Mobile Networks*, ser. WiSec '15. New York, NY, USA: ACM, 2015, pp. 8:1–8:6. [Online]. Available: http://doi.acm.org/10.1145/2766498.2766517

[12] M. Handte, M. U. Iqbal, S. Wagner, W. Apolinarski, P. J. Marron, E. M. M. Navarro, S. Martinez, S. I. Barthelemy, and m. G. Fernandez, "Crowd density estimation for public transport vehicles," in *Workshop Proceedings of the EDBT/ICDT 2014 Joint Conference*, ser. EDBT/ICDT 2014, 2014, pp. 315–322.

[13] M. Handte, M. U. Iqbal, S. Wagner, W. Apolinarski, P. J. Marron, E. M. M. Navarro, S. Martinez, S. I. Barthelemy, and M. G. Fernandez, "Estimating crowd densities and pedestrian flows using wi-fi and bluetooth." in *Proceedings of the 11th International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services.*, ser. MOBIQUITOUS 2014, 2014, pp. 171–177.

[14] A. Janecek, D. Valerio, K. A. Hummel, F. Ricciato, and H. Hlavacs, "The cellular network as a sensor: From mobile phone data to real-time road traffic monitoring," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 5, pp. 2551–2572, Oct 2015.

[15] V. Kostakos, T. Camacho, and C. Mantero, "Towards proximity-based passenger sensing on public transport buses," *Personal and Ubiquitous Computing*, vol. 17, no. 8, pp. 1807–1816, 2013. [Online]. Available: http://dx.doi.org/10.1007/s00779-013-0652-4

[16] L. Mikkelsen, R. Buchakchiev, T. Madsen, and H. P. Schwefel, "Public transport occupancy estimation using wlan probing," in *2016 8th International Workshop on Resilient Networks Design and Modeling (RNDM)*, Sept 2016, pp. 302–308.

[17] NationMaster.com, "Countries compared by media, telephones, mobile cellular, per capita. international statistics at nationmaster.com, cia world factbooks 18 december 2003 to 28 march 2011. aggregates compiled by nationmaster.(assessed 1996-2010)," March 2017. [Online]. Available: http://www.nationmaster.com/country-info/stats/Media/Telephones/Mobile-cellular/Per-capita

[18] J. J. Nielsen, R. L. Olsen, T. K. Madsen, B. Uguen, and H. Schwefel, "Location-quality-aware policy optimisation for relay selection in mobile networks," *CoRR*, vol. abs/1505.01630, 2015. [Online]. Available: http://arxiv.org/abs/1505.01630

[19] S. Schmid, A. Bröring, D. Kramer, S. Käbisch, A. Zappa, M. Lorenz, Y. Wang, A. Rausch, and L. Gioppo, "An architecture for interoperable iot ecosystems," in

2016 2nd International Workshop on Interoperability & Open Source Solutions for the Internet of Things, 2016.

[20] A. Shawky, R. Olsen, J. Pedersen, and H. Schwefel, "Network aware dynamic context subscription management," *Computer Networks*, vol. 58, pp. 239 – 253, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128613003629

[21] J. Weppner and P. Lukowicz, "Bluetooth based collaborative crowd density estimation with mobile phones," in *Proceedings of IEEE International Conference on Pervasive Computing and Communications*, ser. PerCom 2013, 2013, pp. 193–200.

Paper F

Intelligent Parking Assistant - A Showcase of the MOBiNET Platform Functionalities

Lars M. Mikkelsen, Raphael Toledo, and Niels Agerholm

The paper has been published in the
Proceedings of the 22nd ITS World Congress, 2015.

Abstract

The Intelligent Parking Assistant (IPA) system helps the user chose, find and pay for parking. IPA automatically fetches parking lot information from a relevant service, automatically assesses if the vehicle is stopped, evaluates if stopped inside a payed parking lot, and automatically initiates payment for the parking. IPA is published via the MOBiNET platform: A European-wide ITS platform offering functionalities enabling easy migration of services. Migration is enabled by defining a common methodology for publishing services, which covers interfaces, data formats used, coverage area of the service and other. IPA is implemented as proof of concept utilizing MOBiNET as a platform integration use case demonstration. The performance of IPA is evaluated in terms of correctness of decisions made during runtime and based on user input in a trial period, involving 40+ test users using the system on a regular basis.

F.1 Introduction

Today the technology has evolved and it allows users to always be connected and always having a powerful and very capable device with them, namely the smartphone. The constant connectivity of mobile devices allows very high mobility of the user, which means that the services used also must support mobility, and not just mobility within a city but across countries. One particular area related to mobility is Intelligent Transportation Systems (ITS), offering services related to transportation. It is within this sub-area that the Intelligent Parking Assistant (IPA) system has been developed. The IPA system is an in-car parking guidance system that intelligently supports the users in connection with parking, more specifically in finding and selecting parking lots, and paying for parking sessions. The IPA system has been developed taking advantage of the MOBiNET [1] platform and the functionalities this provides, ensuring that the system easily can be migrated to cities all over Europe. The MOBiNET platform aims at acting as a broker of information and services related to transportation. This is done by offering a number of functionalities to support business-to-business (B2B) and business-to-consumer (B2C) services. The services are not hosted on the platform but the platform contains information about services including pricing, coverage area, endpoint, protocol, input, output, and other [8].

Previous approaches to developing an application aimed at helping users finding and selecting parking lots and paying for parking include "SmartPark Trondheim Parkering" [5], "Parker, Find available parking" [3], "ParkMe Parking" [4], and other. Each of these offer a subset of the following: live information about availability; turn-by-turn route guidance; price information; payment in various forms; find your parked car; and other functionalities.

However, where most of the existing systems do well in offering various functionalities, they are limited to operating within certain cities and areas. This could be explained with it being very demanding and cumbersome to obtain the needed information from different locations, in different languages, different pricing schemes, make agreements with local parking companies, ensure that the users can pay even though they cross country and currency borders. These are just some of the obstacles met when migrating services across countries.

In the present effort the IPA system is developed and it is described how it takes advantage of the MOBiNET platform which offers a set of very useful functionalities for developing and publishing ITS services in relation to cross-country usage. The IPA system is to a large extent based on the experiences and development made in the Danish ITS Platform project [9], which was carried out 2010-2013. This project was a large-scale field operation test with GNSS-based on board units (OBU). It consisted of a back-end server, OBUs in 425 cars, and four applications; driver log; dynamic traffic information; traffic statistics tool for road operators; and full-automatic parking payment system for the users. Around 20 million km of driving data has been collected as part of the project. The full-automatic parking payment system was only in operation for a few months. Even though it showed convincing results, much more could be experienced regarding reliability, users attitude and acceptance, and experiences from the parking attendees, just to mention a few topics.

The rest of the paper is structured as follows: Section F.2 gives an overview of the components in the IPA system and the components used by it is given; Section F.3 describes the functionalities of the MOBiNET platform most relevant to the IPA system; Section F.4 describes the technical aspects and functionalities of the IPA system; Section F.5 describes use cases of the IPA system; And Section F.6 describes how the performance of the IPA system is evaluated.

F.2 Intelligent Parking Assistant (IPA) System Overview

The IPA system is made as a very modular system, enabling easy addition of supported areas or regions, by adding services via the Service Directory in the MOBiNET platform. The goal of the IPA system is to help users find and select parking lots based on user input and live information about availability, guide users to the parking lot, intelligently evaluate if the user is inside parking lot when stopped, and automatically initiate a parking session and pay for it.

The IPA system consists of the following components: Park Assist Android app which acts as the main user interaction interface; the Parking Manager service which contains two sub services: Parking Information Provider and Parking Session Manager; the Map Matching server providing map matching functionalities for the application; and the MOBiNET platform that acts as orchestrator and a platform for migration assistance, and where the system is published through. This is illustrated in Figure F.1.

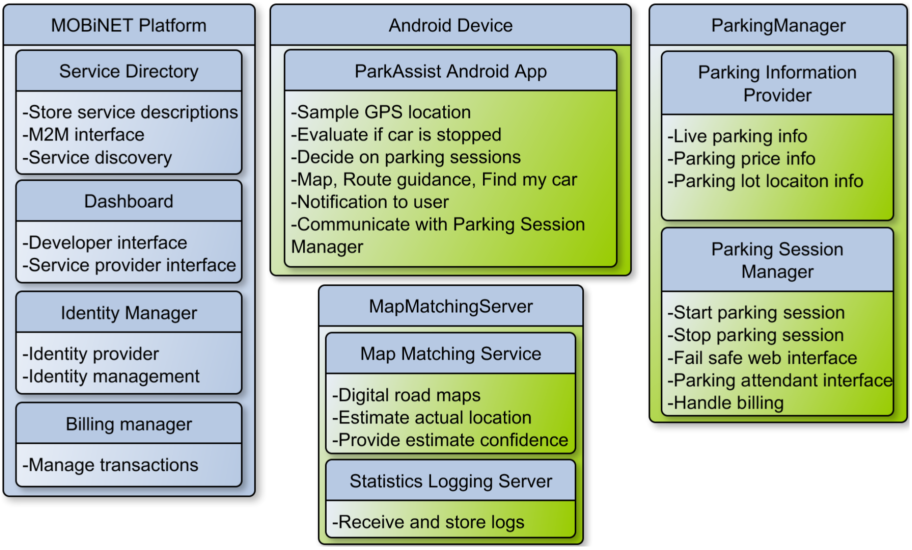


Fig. F.1: IPA System Overview

With this system architecture the ParkAssist Android app is in the centre of operation, as it is installed on the end user device and follows the user wherever he goes. The MOBiNET platform is the orchestrator that depending on where the user is geographically, provides the ParkAssist app with locations of the relevant back end services. That is, the Parking Manager and the Map Matching server are interchangeable and can vary from city to city and country to country.

F.3 MOBiNET platform functionalities

MOBiNET is a project aimed at making Internet of mobility for ITS services of Europe. The idea is to make a platform the one-stop shop for all services related to ITS in Europe. The services include information provider services, front-end services, back-end services, data format translator services, billing and identity management, and other.

MOBiNET is realized via the MOBiNET platform that offers support to B2B and B2C interactions. In Figure F.2 the architecture of the MOBiNET

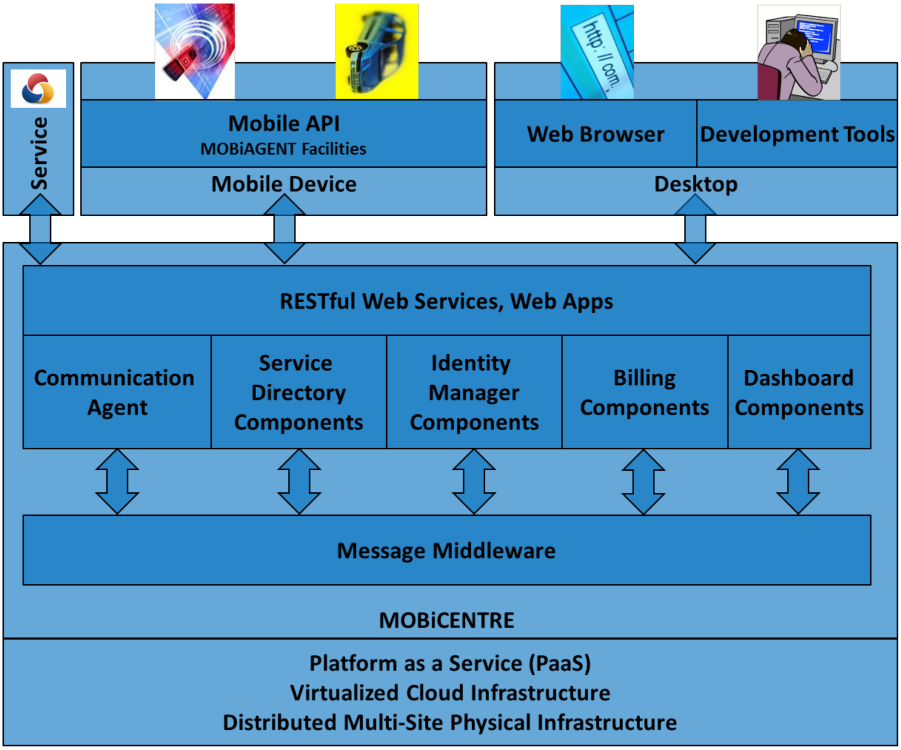


Fig. F.2: The architecture of the MOBiNET Platform [8].

platform can be seen where the key components are: Dashboard, Service Directory, Identity Manager, Billing Manager, Software Development Kit (SDK), and MOBiAGENT. The Dashboard offers access to most of the other components for developers and service providers. The Service Directory contains metadata about services in terms of endpoint, interface type, input and output data types, ownership, pricing scheme, and many other. The Identity Manager handles the identities of both business users and private users, and the component is used both in connection with the Dashboard and by services. The Billing Manager offers payment handling for transactions for B2B and B2C service use. The SDK offers a number of support tools to help developers generate and publish services in the platform. The MOBiAGENT is the client front end software, or app, that users can install on their devices through which interface to searching for services are offered. The MOBiAGENT also offers a number of interfaces for services to use, e.g. to a communication agent for vehicle to infrastructure communication.

In Figure F.3 the main flow and interactions between Dashboard and Service Directory can be seen, along with how end user services interact with the Service Directory.

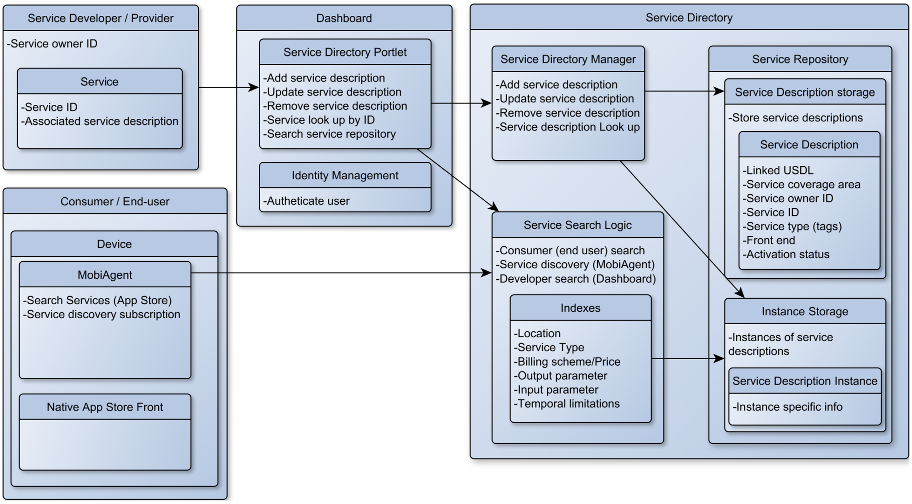


Fig. F.3: Dashboard and Service Directory interactions [7].

In the scenario of the IPA system the components of the MOBiNET platform are used in the following way.

- Service description of the front-end service (app) ParkAssist is added to the Service Directory so end users can find and download the app.
- Service descriptions are added to the Service Directory of the back-end service providing parking information and parking session management, and of the Map Matching server.
- The back-end services are discovered by the app by contacting the Service Directory during runtime and querying services based on service type and the location of the user device.
- The Identity Manager is used to verify the user credentials, and to link a user with a parking bill.
- The Billing Manager ensures that transactions are performed from the user to the parking service provider.

F.4 Intelligent Parking Assistant System

F.4.1 App Functionalities

The IPA app, ParkAssist, has a number of functionalities, both in terms of communication with back-end services and user interaction. In Figure F.4 an overview of the layout of the ParkAssist app can be seen.

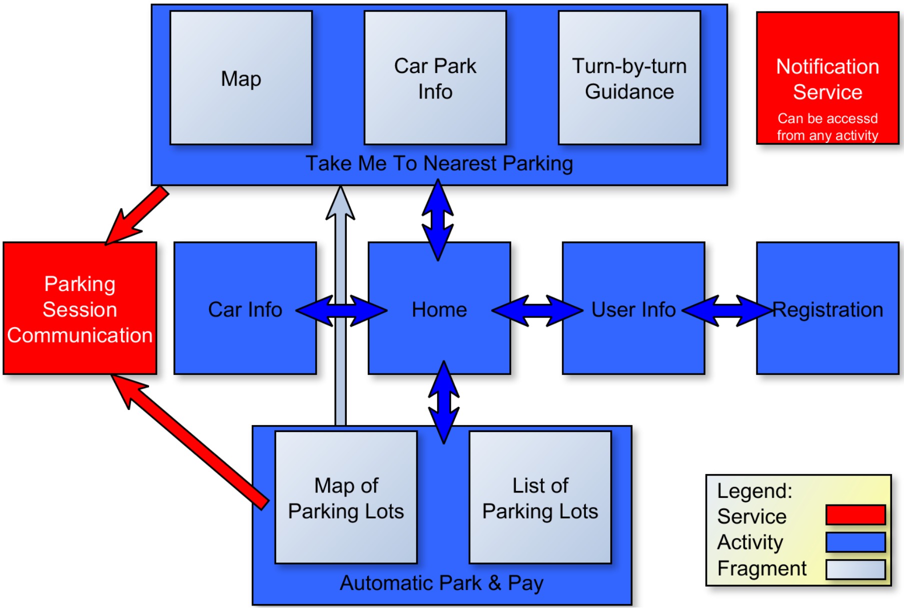


Fig. F.4: ParkAssist app overview. Service: Background functionalities such as communication handling. Activity: Foreground functionalities including user interactions. Fragment: A view of the activity, where there can be more views for one activity that can be navigated for instance by swiping [2].

The ParkAssist app runs under some assumptions about the context, that is needed for proper interaction with the user, and to be able to make correct decisions. It is assumed that the device is fixed within the vehicle until at least some seconds after the vehicle is stopped. It is also assumed that the screen is on and visible to the user while searching for an available parking spot and parking the car. The main interactions with the app for the user are:

- Automatic Park & Pay Presents all the parking lots to the user, from which he can choose to be guided to or just park.
- Take Me To Nearest Parking Proposes the nearest parking lot to the user, which the user can choose to be guided to.
- Car Info Allows the user to add and store cars and information about them.

- User Info Where the user can add personal information, register and log in to the system.

When either Automatic Park & Pay or Take Me To Nearest Parking is chosen, the app will initiate automatic payment whenever the car is stopped within a parking lot. In order to be able to use the automatic payment functionality the user need to identify himself in terms of a user id and password, and enter the license plate of the vehicle used. This is a one time operation performed by the user. This information are used for identification with the Parking Session Manager service, to let it know which user to bill for the parking sessions. The license plate is used in connection with starting and stopping parking sessions at the parking session manager service. In this way it is possible for a user to use the system with multiple vehicles but with the same user credentials. For this reason the app contains two pages where the user can input and manage information. A page for user identification and information, and another page for car information and details. In the latter page it is possible to see the location of the car, based on information the user has provided about the car. This will also be used in connection with parking where the GNSS location of where the car is parked will be saved.

One of the key back-end communication functionalities of the app is that it communicates with a Parking Information Provider service and fetches the available parking lot information from the current relevant parking information service. The Parking Information Provider service is chosen by making a service discovery query to the Service Directory in the MOBiNET platform. The query includes service type and the location of the device to match service coverage area. If there is no match the search area is expanded. If a service matches the discovery query it is queried for live availability information of the parking lots covered by the service. The information is fetched from the service when initially discovered when the app is opened, and periodically hereafter when the map is open in the app and when any kind of search for parking is performed.

A parking session is automatically initiated with the Parking Session Manager if the car is evaluated to be stopped, at which point the current location is compared to the area of the parking lot. Whether the car is stopped is evaluated based on the GNSS location samples and information about velocity. The location is sampled and if two sequential samples are sufficiently close, and if the speed is sufficiently low, the car is assumed to be stopped. Due to the GNSS location having an accuracy of some meters, it is possible that the location samples will vary even though the device is actually not moving. The same is the case for the speed. Because of this uncertainty the user also has the option to manually let the app know that the car is stopped, after which the intelligent parking session initiation will run as normal.

The inaccuracy of the location samples is a big issue, as the user could end up paying for parking sessions that never took place. The location inaccuracy is therefor counteracted by using the Map Matching service. Map matching is estimating the true location of location samples with inaccuracies by including digital maps of roads where the locations are matched to. For more information about the map matching service see the description in Section F.4.2. The estimated true location of the device that is returned from the Map Matching service is used to evaluate whether the vehicle is within a parking lot that requires payment, or if the vehicle is parked elsewhere.

The app implements a backup solution to the map matching, which simply checks the last of the non-corrected location samples for whether the vehicle is within the parking lot area. This is both in case the Map Matching service is unavailable, but also in case of the Map Matching service not having the maps of the current area available. The latter is possible because the Map Matching service only is optional in order for the system to function, as opposed to the Parking Manager service which is mandatory as it contains parking availability and pricing information.

If the car is estimated to be stopped within a parking lot where payment is required and supported by the Parking Session Manager, the parking session is started automatically. If this is done a notification is shown to the user. This notification has two sequential states. In the first state there is an initial grace period of a few minutes where the user can cancel the parking session without having to pay for anything. This is to ensure that the user has the final saying whether he want to start a parking session and ultimately pay for parking. In the second state the user has the option to stop the parking session, and at this point pay for the parking session. The notification is persistent such that the user is forced to interact with it to make it go away, i.e. cancel or stop the parking session. In the case where the notification for some reason disappears, for instance due to an app crash or other, the user has the possibility to access the Parking Session Manager via a web interface. The web interface requires login with user id and password, which is the same credentials used in the app. The web interface allows the user to see all his parking sessions, and to manually end active parking sessions.

F.4.2 Back-end Services and Parking Lot Information

Besides the application the core component of the IPA system is the Parking Manager service. This service has 2 main functionalities: providing information about parking lots, and managing parking sessions. The Parking Manager service is illustrated in Figure F.5.

The information providing functionality of the service offers information about parking lots in terms of location, capacity, live availability, pricing, and other information. This information is used in the app to provide the user with information about parking lots in the area, and to help him select a

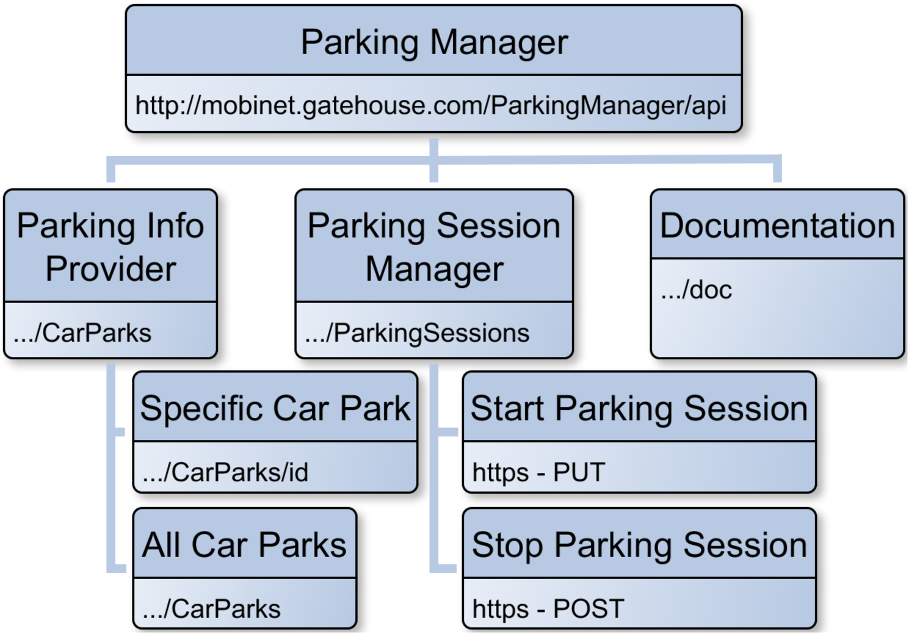


Fig. F.5: Parking Manager service overview.

parking lot to use. This is provided via a Representational State Transfer (RESTful) interface where the information is formatted as JavaScript Object Notation (JSON) for easy parsing and processing.

The parking session management functionality of the service provides an interface to start new parking sessions and to stop active parking sessions. This functionality is also accessible via a web interface, where users can log in and see all their parking sessions, and as a fail safe stop active parking sessions. When a parking session is stopped the related user is automatically billed for the parking, by the Parking Session Manager interacting with a billing service.

The Map Matching service is provided on a server as a web service. The service provides an interface to submit an array of GNSS location samples, including speed, accuracy, and orientation information. The map matching is done using an algorithm that is developed and described in [6]. The algorithm takes location samples as input, which are matched with digital maps of the corresponding area. The digital maps are stored at the server, and contains machine interpretable information about the road sections and intersections. Based on the input locations and the attached information the map matching algorithm returns estimates of the actual locations, and a confidence in the updated estimates. The map matching step is not mandatory for the IPA to function, but it adds greater confidence to the decision of initiating parking sessions automatically.

F.5 Use Case

There are two main use cases of the IPA system for the end user. The first use case is as follows: A user drives to work regularly in a big city, and each day when he approaches the city he opens the ParkAssist app on his smartphone and initiates the Automatic Park & Pay functionality. He knows where the parking lot is so he navigates to it, parks, and leaves the car with his smartphone. Once he stops his car the app transmits a number of GNSS locations to the Map Matching service. The location logging is initiated as soon as the car approaches the parking lot and stopped when the car is parked. The Map Matching service replies with the GNSS locations adjusted according to the digital road map of the area. This reply is used in the app to evaluate if the car is parked outside or inside the parking lot. If it is estimated that the car is parked inside the parking lot a parking session is initiated with the Parking Session Manager. When the user is ready to go to his car and leave he opens the app and selects his car, and is now presented with directions to where he parked his car. When he reaches his car he stops the parking session, by interacting with the persistent notification on his smartphone, and the payment is automatically performed from his account to the parking company.

The other use case is more oriented at showing the migration of the system, and is as follows: The same user as described above, is going on a holiday to another city in another country. When he approaches the city, he opens the app on his smartphone and searches for the attraction that he is going to visit. The app now proposes the parking lot nearest to the attraction, and is presented with information related to the parking lot such as the number of available slots and the pricing scheme. The user selects the parking lot and receives turn-by-turn route guidance to the parking lot. When he arrives at the parking lot the same automatic sequence, as described in the first use case, is done of starting a parking session and paying when done. At the time when the user opened the app when he was initially approaching the city in the new country, the app contacted the MOBiNET platform, more specifically the Service Directory, and discovered the Parking Info Provider service relevant for the new city. When the car was stopped the Map Matching server was contacted to check if the new city is supported by digital road maps. If it is not supported the fall-back method is used where the most recent location sample is evaluated locally on the device along with the parking lot area.

From the two scenarios it can be seen how the process of finding a parking lot and paying for it can be done just as simple in a new town as in a town the user knows well. In both scenarios the user saves time spent driving around looking for parking lots, and thereby reduces congestion on the roads and reduces CO2 emission. The user also saves some time that is spent on paying for the parking session and he does not need to learn and understand a new payment methodology for each parking lot.

F.6 Evaluation of the System

F.6.1 Performance Indicators

The IPA system will be evaluated based on two types of parameters; user related, and accuracy of decision.

The user related parameters include how the user perceives the system and its functionalities, meaning to check if the system does what the users expect and does it properly, and to understand if the users think that the system actually makes the task of parking easier. Also statistics of usage of the system and its functionalities will be evaluated, to see if the functionalities of the system are actually used and how much.

The accuracy of decision parameters include how accurate the decision of initiating a parking payment is, both in terms of when the decision is made, if the decision is correct, and if no decision is made. Furthermore, as the decisions rely heavily on the GNSS location, and on having an accurate estimate of it, the location samples are also evaluated manually to check if they are correct and make sense, and to understand the general quality of the samples.

F.6.2 Evaluation Scenario

The IPA system will be evaluated based on a test performed in Aalborg in the northern part of Denmark, where approximately 40 users will be part of a test, where they will be using the application when searching for and parking on parking lots, and when paying for the parking sessions. In Aalborg 11 parking lots with time-based parking payment will be a part of the test and the cars of the users will be equipped with a sticker in the windshield that parking attendants can scan to check if the car in question currently has an active parking session, or if the car should be fined. This evaluation scenario is what is described in the first use case described in Section F.5. Later on it is planned that additional cities across Europe will implement a Parking Manager service, effectively realizing the second use case described in Section F.5.

F.7 Next Steps

Currently, besides the live test with real users, the IPA system is in the process of being migrated to Trikala Greece. To make such a migration possible a local parking information service is made available. This is the only mandatory step for a migration. For the rest of the IPA system the existing components can be used for the Greek users without modification. However, additionally both the app and the web interface are translated to Greek for convenience. The reason for so few steps being required to perform a migration of the system to a city in another county is the MOBiNET platform. Key functionalities offered by MOBiNET in this context are identity management, billing management, and discovery of services on the go based on location and output data format. These functionalities allow for easy reusability of software system components in other contexts.

Another aspect, however much more technical, is evaluating the impact of poor network conditions on the performance of the system. Particularly the performance of the network while the map matching is being performed is of interest. The map matching process consists of collecting location samples, transmitting the samples from device to server, running the map matching algorithm, transmitting the matched location samples from server to device, and evaluating the matched locations on the device. If the network performance is poor during the transmissions of the samples back and forth, the user might have moved before the reply arrives at the device. This could lead to a wrong decision based on the reply, or that the reply is never received, why no decision can be made. In either case it would be preferable to use the backup solution and evaluate locally if the device is within the parking lot or not, despite the lower accuracy of the data to base the decision on. For this reason, the performance of the connection is an important factor in performance of the system.

References

[1] "MOBiNET - Internet of Mobility," November 2012. [Online]. Available: http://mobinet.eu/

[2] "Android developer," December 2014. [Online]. Available: http://developer.android.com/index.html

[3] "Parker, find available parking," November 2014. [Online]. Available: https://play.google.com/store/apps/details?id=com.streetline.parker

[4] "Parkme parking," December 2014. [Online]. Available: https://play.google.com/store/apps/details?id=com.parkme.consumer

[5] "Smartpark trondheim parkering," November 2014. [Online]. Available: https://play.google.com/store/apps/details?id=no.wtw.tp

[6] N. Agerholm, H. Lahrmann, B. Jørgensen, A. K. Simonsen, and S. Tøfting, "Full-automatic parking registration and payment - in principle GNSS-based road pricing," *10th ITS European Congress*, June 2014.

[7] L. Mikkelsen, R. Olsen, F. Alesiani, S. Longo, T. Schlauch, and A. Rickling, "Deliverable 3.4a Service directory concept (release 1)," MOBiNET - Internet of Mobility, Tech. Rep., July 2014. [Online]. Available: http://mobinet.eu/

[8] T. Schlauch, D. Beckmann, I. Passchier, N. Zahariev, and L. Mikkelsen, "MOBiNET - an innovative approach for a European-wide ITS service platform," *10th ITS European Congress*, June 2014.

[9] S. Tøfting, H. Lahrmann, N. Agerholm, B. Jørgensen, and L. V. Johannsen, "ITS Platform - development of intelligent traffic," *10th ITS European Congress*, June 2014.

ISSN (online): 2446-1628
ISBN (online): 978-87-7112-938-0

AALBORG **UNIVERSITY PRESS**